I came up with a very simple algorithm that decomposes a 2D curvilinear GPS generated racetrack into multiple straight segments. The user has the capability to control the number of straight segments by adjusting the delta heading allowed between straight segments. The smaller the tolerance the more segments are generated. The MATLAB code is provided in Appendix A. I ran this algorithm against a partial data set of the Zumwalt profile and the test results looked very promising. A more detailed statistical analysis is needed to evaluate the accuracy of this algorithm. The analysis is not simple since we are dealing with data vectors of different sizes. That will be the next step. I also encourage the reader to compare the performance of my algorithm against other algorithms he found on the Web. I realize that there are a lot of ways to skin a cat. The best algorithm will be the one that minimizes the overall Cross Track error between the generated pattern and the real GPS path using the lowest number of straight segments.

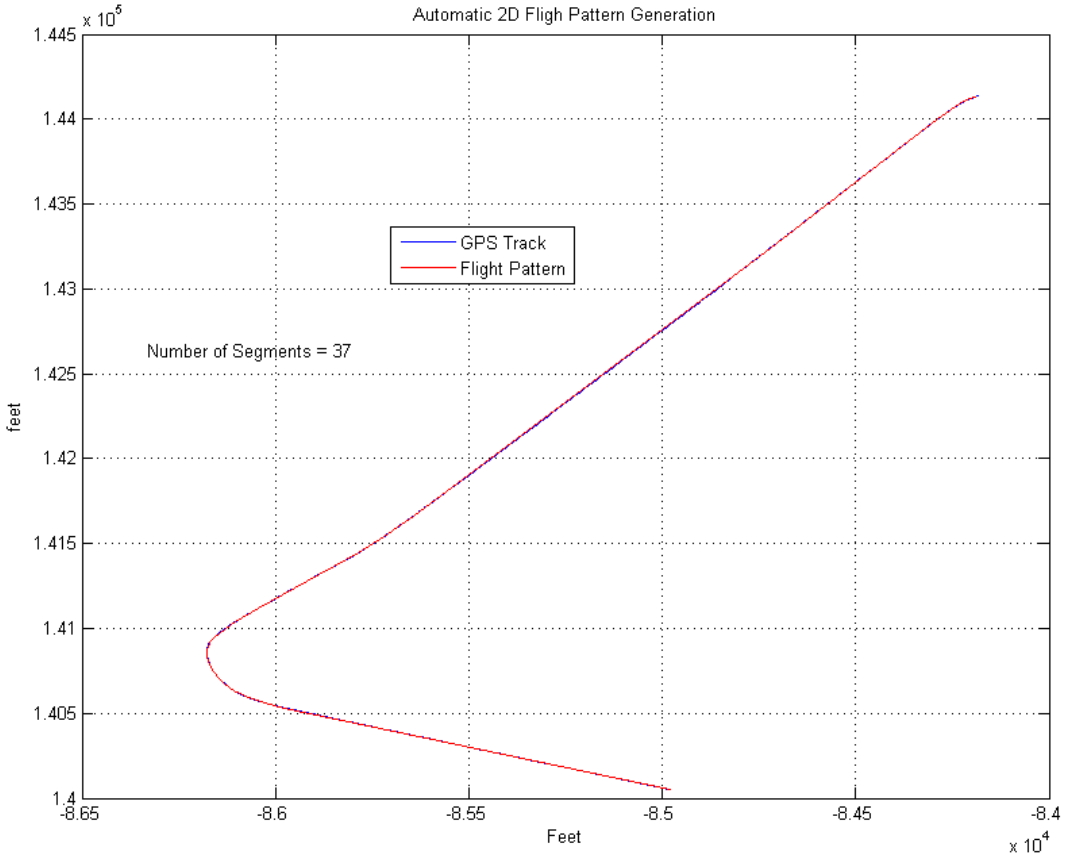

Figure 1 Ground Track vs. Flight Pattern

The following data shows the coordinates of the generated straight segments

Way point Generator

Segment No.	X(ft)	Y(ft)
1	-84979	140049
2	-84980	140050
3	-85978	140533
4	-86027	140561
5	-86057	140582
6	-86080	140601
7	-86099	140623
8	-86104	140629
9	-86121	140655
10	-86131	140674
11	-86145	140704
12	-86159	140742
13	-86167	140776
14	-86173	140805
15	-86176	140834
16	-86177	140876
17	-86175	140892
18	-86172	140912
19	-86169	140923
20	-86159	140946
21	-86128	141002
22	-86104	141039
23	-86045	141117
24	-85887	141320
25	-85849	141365
26	-85785	141445
27	-85729	141527
28	-85653	141646
29	-84736	143214
30	-84302	143975
31	-84244	144071
32	-84231	144089
33	-84224	144098
34	-84218	144106
35	-84205	144119
36	-84196	144127
37	-84190	144130

APPENDIX A – MATLAB CODE

```
%-------------------------------------------------------------------------
%  A very simple algorithm to map a ground track using straight segments
%-------------------------------------------------------------------------

clear all
load zumwaltxy1.txt;                    % Load Partial Zumwalt Pattern Data
q=1:1000;                               % Define points related to one way trip
nx1=zumwaltxy1(q,1);                    % Load GPS Topocenric X-coordinate in feet
ny1=zumwaltxy1(q,2);                    % Load GPS Topocenric Y-coordinate in feet
dt=0.1;                                 % Delta time in seconds between GPS sample
points
taua=0.5;

[nx datar1] = lag(dt,taua,nx1);         % Smooth out GPS data with a 0.5 second time
constant low pass filter
[ny datar2] = lag(dt,taua,ny1);

%---------------------------------
r2d=180/pi;
deltahdg = 2.5;                         %Heading tolerance between segments in degrees
nseg=250;                               % Maximun number of straight segments allowed
%---------------------------------
np=length(nx);
p=1;
wpx=zeros(1,nseg);
wpy=wpx;
wpx(p) = nx(1);
wpy(p) = ny(1);
j=1;
k=2;

 while ( k < np & p < nseg )
    dhdg = atan2((ny(k)-ny(j)),(nx(k)-nx(j))) * r2d;        %compute the heading in
degrees between nk and nj
    prevhdg = atan2((ny(k)-ny(k-1)),(nx(k)-nx(k-1))) * r2d; %compute heading in
degrees for last segment
    if( abs(dhdg - prevhdg) > deltahdg )                    %check for heading delta
tolerance
       % Start a new straight line
        j = k - 1;
        p = p + 1;
        wpx(p) = nx(j);                                     %Store x-coordinate of
new straigth segment
        wpy(p) = ny(j);                                     %Store y-coordinate of
new straigth segment
    else
        % Pickup next way point and continue with the same straight line
        k = k + 1;
    end
```