Optimized Processing of Satellite Signal Via Evolutionary Search Algorithm

Azmi Hassan
Satellite Navigation Research Group (SNAG)
Centre For Technology Policy & Int. Studies
Universiti Teknologi Malaysia
81310 UTM Skudai
email: azmi@fksg.utm.my

Rusli Othman & Tang Kieh Ming
Department of Geomatic Engineering
Faculty of Engineering & Geoinformation
Universiti Teknologi Malaysia
81310 UTM Skudai
email: rusli@fksg.utm.my

Abstract

Researchers from the Satellite Navigation Research Group (SNAG) of UTM are currently conducting a research program that mitigates the effect of the Anti Spoofing (AS) policy. A robust strategy, called the Pseudo Randomized Search Strategy (PRSS) has been developed to counter the effect of this AS policy. The PRSS algorithm is an adaptive search technique that can learn high performance knowledge structure in reactive environments that provide information as an objective function. A combination of three methods, namely optimization, global random search and ambiguity function mapping has produced an efficient and robust mitigation technique. Numerical results indicate that, in all the test cases, no more than 4% search of the total search space was investigated to determine the correct set of answers. This result implies that the size of the search window does not play an important role in determining the search efficiency. This will take away the constraint resulted from the AS policy in processing the GPS satellites signal. The algorithm also shows its robustness because it does not require a good initial starting point. Using different initial point, all of them produced the correct results.

Keywords
Satellite, optimization, global random search.

I. INTRODUCTION

Couselman and Gourevitch (1981) introduced the Ambiguity Function Mapping (AFM) technique in their paper titled "Miniature Interferometer Terminals for Earth Surveying : Ambiguity and Multipath with the Global Positioning System". The root of the AFM is believed to be derived from VLBI techniques. Remondi (1984) first used this method extensively for GPS static positioning and later also for pseudo-kinematic positioning (Remondi, 1989). Mader (1990, 1992) also used the AFM for rapid static and kinematic GPS positioning. The most recent use of this method was by Han (1996), who gained some improvement on the computation time, particularly on the grid step size used. But beyond this, the AFM gained little popularity.

Theoretically, as proven by Lachapelle, et al (1992b), the AFM is equivalent to the Least Squares Search method widely used in other search techniques such as Fast Ambiguity Resolution Approach (Frei and Beutler, 1990), Cholesky Decomposition method (Landau and Euler, 1992), Least Squares Search (Hatch, 1990) and, most recently, the Least Squares Ambiguity Decorrelation Adjustment (Teunissen, 1994). Most of the above techniques have been incorporated in commercial GPS processing software packages.

One of the main features of the AFM compared to other techniques is that it is immune to cycle slips. Although most researchers have shied away from the AFM, in fact the AFM is the first on-the-fly ambiguity resolution technique introduced. The main reason for the lack of the popularity of the AFM is largely its computational burden. To reduce the computation burden of the AFM, the most obvious step is to reduce the mathematical operations needed to determine the position that produces the maximum ambiguity function. Since AFM works in the position domain, good initial coordinates of the unknown point are needed in order to establish the search window. Another point that is problematic is that the grid for the search step size needs to be determined beforehand. If the step size is too small, then the computation will take a long time and, if it is too coarse, the correct position may possibly be eliminated. Han (1996) reported that using a certain combination of L_1 and L_2 frequencies, the search area should be within $\pm\ \lambda$ for six satellites and the step size should be less that one-tenth of the observable wavelength. This method works best with dual frequency receivers since they can provide good initial position estimates by reducing the effects of the ionosphere. But, with only L_1 measurements available, the method requires so many measurements that it is rendered useless for kinematic applications.

This paper will address the combination of optimization, global random search strategy and AFM to produce a highly efficient and robust technique in resolving the ambiguity on-the-fly using only L_1 measurements. A brief review on each of the three strategies will be discussed. It will be followed by the combination strategy used to produce a highly efficient search algorithm and finally numerical results are presented.

II. AMBIGUITY FUNCTION MAPPING

The AFM uses a function to determine the maximum value of a certain position. The function used in practice is as follows:

$$A(x_o, y_o, z_o) = \sum_{i=1}^{m}\sum_{j=1}^{n}\cos\left\{2\pi\left[\phi_{obs}^{mn}(x,y,z) - \phi_{com}^{mn}(x_o, y_o, z_o)\right]\right\}$$

where $\phi_{obs}^{mn}(x,y,z)$ is the double difference observed phase at the true position (x,y,z) and $\phi_{com}^{mn}(x_o,y_o,z_o)$ is the computed double difference phase at the trial position of (x_o,y_o,z_o). The summations over i and j refer to the total number of epochs m and number of satellites n. The function A will reach a maximum value when the trial position (x_o,y_o,z_o) is equal to the true position (x,y,z). In the case of one epoch, m=1, and one satellite, n=1, and assuming there are no biases or errors, the maximum of A is 1. As the trial position (x_o,y_o,z_o) is varied within a volume of the search space, a pattern of maxima and minima will be observed at each trial position and the correct position will be identified as a peak. If enough measurements are available, this peak is distinguishable among other relative maxima.

The AFM works by trial and error within the search area space. For example, if the search space is 1 m x 1 m x 1 m and the step size used for the trial position is 1 cm, for m satellites and n epochs there will a total of $(m*n*100^3)$ trial positions to be tested. Clearly, for this method to work efficiently, a good initial position is needed so that a small search area can be constructed. If the method of trial and error is used, then dual frequency measurements will have shorter computation times compared to single frequency because of a wider lane.

In the traditional AFM, the function is computed at every grid point within the search volume in an attempt to find a unique set of integer numbers. Using the traditional AFM method to develop the objective function, there will result three unknowns, that is, the coordinates (x,y,z) of the position. In this research, to avoid estimating an initial position and extra unknowns, a direct search for the integer numbers is performed instead. In this case, rather than an initial position being defined, an initial set of integer numbers is defined.

Therefore, the previous function is rewritten as:

$$A(\hat{n}_i) = \sum_{i=1}^{m}\sum_{j=1}^{n}\cos\left\{2\pi\left[\phi_{obs}^{mn}(n_i) - \phi_{com}^{mn}(\hat{n}_i)\right]\right\}$$

where, instead of position (x,y,z) as unknowns, the modified equation uses the carrier phase ambiguities (n) as the unknowns.

III. SEARCH AND OPTIMIZATION

Optimization theories encompass the quantitative study of optima and methods for finding them. When an optimization process is performed, it can be said that we are seeking to improve the performance toward some optimal point or points. The method used to derive the optimum point is called search technique.

Basically, there are three types of search techniques: calculus-based, enumerative and random. The calculus-based method has been used very extensively and can be categorized into indirect and direct techniques. Indirect techniques seek local extrema by solving the nonlinear set of equations resulting from setting the gradient of the objective function to zero. On the other hand, the direct searches seek the local optima by hopping on the function and moving in a direction related to the local gradient. The enumerative scheme is a straight-forward search method that begins in a finite search area and the search algorithm starts looking at objective function values at every point in the space. Lastly, the random search method uses probabilistic methodology to guide the search for an optimum point.

The calculus-based method, if used together with AFM, does not provide any real advantage. The reason is that this method is very local in nature. The optima sought are the best in the neighborhood of the current point. This means that the GPS phase measurements must have a very low noise or biases. For a single frequency measurement, this assumption is improbable since it is corrupted with multipath and ionospheric delay errors. Secondly, this method depends heavily on the existence of derivatives (well-defined slope values) and this requirement adds another burden to AFM since computation of derivatives is very expensive.

The current AFM uses the enumerative search scheme. Using this technique, improvements in computation time have been made, for example by Remondi (1991), Mader (1992) and Han (1996). The enumerative search works well under conditions of dual frequency and good initial trial position but does not work well for single frequency measurements corrupted with multipath. This led to the choice of the random search method in the AFM.

There are various methods of random search that can be used to improve the robustness and efficiency of the AFM. But one method that stood out among the rest is based evolutionary algorithms. This particular method is based on the collective learning process within a population of individuals, each of which represents a search point in the space of potential solutions. One particular point that interested the authors is that the population can be arbitrarily initialized. What this means is that, if this method is applied to AFM, the trial ambiguity can be initialized arbitrarily and the search space does not need to be defined beforehand. This algorithm, coupled by randomization processes of *selection*, *mutation* and *crossover*, evolves toward better and better regions of the search space. The process of *selection* exists both in calculus-based and enumerative searchs but are very deterministic in nature. The processes of *mutation* and *crossover* are exclusive to the evolutionary algorithms. Basically, the environment of the evolutionary algorithm induces quality information (in this case, the ambiguity function value) about the search points, and the *selection* process favors those trial points of higher ambiguity function value to reproduce more often than those of lower function value. The *crossover* process allows the mixing of the main trial ambiguity information while passing it to a new set of trial ambiguities. The process of *mutation* introduces innovation into the population set.

IV. PSEUDO RANDOMIZED SEARCH STRATEGY

A skeleton of the evolutionary search algorithm is shown in Fig. 1 below. During iteration t, the algorithm maintains a set of trial ambiguities $P(t)$ of structure $\{\hat{n}_1^t, \hat{n}_2^t, ...\hat{n}_N^t\}$ where $\hat{n}_1^t$ is the trial ambiguity 1 for iteration t, N is the total number of trial ambiguities used in each iteration and P is the population set. The size of N remains fixed for the duration of the search.

```
t = 0;
initialize P(t);
evaluate A ( n̂_i ) at P(t);
while (not termination condition)
{
    t = t + 1;
    select P(t) from P(t-1);
    crossover P(t);
    mutate P(t);
    evaluate A( n̂_i )at P(t);
}
```

Fig. 1: Evolutionary Search Algorithm

Each ambiguity $\hat{n}_1^t$ is evaluated by computing $A\ (\hat{n}_1^t)$ at $\hat{n}_1^t$. The values of the ambiguity function provide a measure of fitness of the evaluated structure. When each ambiguity in the trial set has been evaluated, a new set of trial ambiguities is formed in three steps. First, the structures in the current iteration are selected to reproduce on the basis of their ambiguity function value. That is, the *selection* algorithm chooses structures for replication by a stochastic procedure that ensures that the expected number of new ambiguities associated with a given structure $\hat{n}_{i1}^t$ is $A(\hat{n}_i^t)/\mu(P,t)$, where $A(\hat{n}_i^t)$ is the ambiguity function value of the structure $\hat{n}_i^t$ and $\mu(P,t)$ is the average performance of all structures in that particular set of trial positions. What this means is that the structures that performed well may be chosen several times for replication and structures that performed poorly may not be chosen at all. Using only this type of procedure as is the case for most other search algorithms, would cause the best performing structures in the initial set of positions to occupy a larger and larger proportion of the trial sets over time. This is where the processes of *crossover* and *mutation* come into play.

Next, the selected structures are combined to form a new set of structures for evaluation using the *crossover* process. This procedure will combine two trial ambiguities, say, n_i^t and n_k^t for sets i and k and iteration t, respectively, to produce new and better ambiguities, n_i^{t+1} and n_k^{t+1}. This process operates by swapping corresponding segments of string representations of ambiguities n_i^t and n_k^t.

In generating new structures for testing, the *crossover* process draws only on information present in the structures of the current iteration set. If specific information is missing, then it is unable to produce new structures that contain it. The *mutation* process, by arbitrarily altering one or more components of a selected structure, provides the means for introducing new information into the position set.

V. OPTIMIZED AMBIGUITY FUNCTION METHOD

The evolutionary search algorithms, as previously described, are used here in the AFM. But, instead of using position as the search parameter, the ambiguity integer number of the carrier phase was used as the parameter to be searched. The main advantage is that, since the search algorithm works on the binary coding (0,1) of the parameter itself, it is easier to work on the ambiguity number than the position itself. For example, consider the three double difference ambiguities for a four satellite configuration. Table 1 shows the coding used in the search algorithm for sets i and k.

Table 1 : Binary Coding

Satellite Pair	Ambiguity	Binary Coding
SV# 2-7:i	431 550	00000011101010010110
:k	431 425	10000011101010010110
SV# 2-15:i	454 520	00011110111101110110
:k	454 524	01101110111101110110
SV# 2-26:i	155 356	10111011011110100100
:k	155 340	10111011011110100100

In Table 1 each set of three ambiguity numbers can be used to derive one initial position. The three ambiguities are considered one string since they are concatenated together. In this paper, a set of eight ambiguities are used for each iteration and, therefore, for each iteration, there will be eight concatenated ambiguity strings.

To see an example of how the process of *crossover* is performed, look at the ambiguities for SV 2-7 of sets i and k in Table 1. In a one-point *crossover* that has been implemented in this search, a point is chosen at random (using a roulette wheel procedure) to swap ambiguities i and k to produce a new set for the next iteration. The process is illustrated in Fig. 2 below.

To make sure that diversities exist and most importantly to prevent a premature convergence of a local optimum, the process of *mutation* in implemented. This process is a random alteration on a particular string of ambiguity where the point chosen will be change from 0 to 1 and vice versa. Fig. 3 shows the process of a one-bit *mutation* of the SV# 2-7 set i ambiguity.

Initial Coding New Coding
set i 0000001*1101010010110 0000001*1101010010110
* is the crossover point
set k 1000001*1101010010110 0000011*1101010010110

Fig. 2 : Crossover Process

0000001 101010010110 0000001 101010010110
Initial Coding New Coding

Fig. 3 : One Bit Mutation

VI. RESULTS AND ANALYSIS

To show the validity of this search algorithm, static data of only L_1 measurements was used. Three data sets were used, one each for a short (3-km), medium (12-km) and long (21-km) baseline. Since the purpose is to validate the search algorithm of the optimized AFM, the results shown are based on the correct set of ambiguity numbers and not the correct position, as is usually done for the AFM.

The 'true' ambiguities for these measurements were determined by processing all available epochs of measurements using the *ASHTECH*™ *GPPS* processing software. The processing results are as shown in Table 2 below.

Table 2 : True Double Difference Ambiguities

Baseline	SV Pair	DDN	Epochs	σ(XYZ)
3 km	2 - 15	454530	495	1.30 cm
12 km	2 - 7	40004	514	0.82 cm
21 km	7 - 14	1906348	488	1.27 cm

Based on the standard deviations (σ) for the positions, it can be safely assumed that the ambiguities obtained are the true values. The test that a search has reached a global optimum will therefore be based on comparisons with the above values.

The PRSS algorithm depends on the values of two main parameters, the probability values of *crossover* (P_c) and *mutation* (P_m). In order to find the best combination of these two parameters, various tests were performed. The primary concern in this research is to minimize the number of measurements used, that is, to use the minimum number of measurement epochs. In this case, measurements of only one epoch were used. The parameter value ranges that were found to give the best results are as shown in Table 3. This particular test numbers was chosen after some preliminary test was performed found that this range work the most efficiently.

Table 3 : Probability Parameters

Parameter	Test Range
P_c	0.5 - 0.9
P_m	0.0001 - 0.001

The maximum value of the ambiguity function was normalized to one. The iterations were stopped when the ambiguity function reached a value greater than 1 and, also, all ambiguity set values equaled the same value.

The first iteration of this search starts with initial ambiguities of 807110, 246954, 132617, 98377, 1008399, 462367, 459438 and 636341, with corresponding fitness values ranging from 0.188069 to 0.270883. The next iteration shows a fast convergence to an optimum value. For example, for ambiguity set 0, there is nearly a 50% change from initial value of 807110 to 462367. Table 4 shows the values of the first, second and third iterations of ambiguity. The same observations can be made for other set ambiguities. It can be observed that, for iteration 2, there exists some pattern of uniformity among the ambiguities set. For example, ambiguity sets 1, 2 and 7 have the same ambiguity value of 462367 and ambiguity set 6 has a value of 464366. This change was achieved through the process of five crossovers and zero mutations. Fig. 4 shows part of the computer output for the first two iterations.

Table 4 : Iteration 1, 2 and 3 of Epoch 1 for the 3 km Baseline

Set	Iteration 1	Iteration 2	Iteration 3
1	807110	462367	462511
2	246954	462367	459295
3	132617	807367	459439
4	98377	246950	462366
5	1008399	459439	462366
6	462367	462366	462367
7	459438	462367	459438
8	636341	459438	462367

Since ambiguity sets 5 and 6 with ambiguity values of 462367 and 459438 respectively, the highest fitness values of 0.256765 and 0.270883, it is expected that the next iteration produced will be confined to these two higher fitness value ambiguities set. This is shown in the next iteration where six out of eight ambiguities produced are from combinations of the two higher fitness value ambiguities. The next iteration shows that the ambiguity values of 462367 (#5) and 459438 (#6) have taken over the whole set of ambiguities.

```
-------------------------------------------------------------------------------
Iteration  0                                                       Iteration  1
num        string          fitness          string                      fitness
-------------------------------------------------------------------------------
1) 01100011000010100011  0.180269 | 11111000011100001110 0.256765
2) 01010101001000111100  0.188069 | 11111000011100001110  0.256765
3) 10010000011000000100  0.181562 | 01010011000010100011  0.180269
4) 10010010000000011000  0.180127 | 01100101001000111100  0.188069
5) 11110000110001101111  0.174112 | 11110101010000001110  0.270876
6) 11111000011100001110  0.256765 | 01111000011100001110  0.256769
7) 01110101010000001110  0.270883 | 11111000011100001110  0.256765
8)10101101101011011001  0.190127 | 01110101010000001110  0.270883
-------------------------------------------------------------------------------
Iteration 0 Accumulated Statistics:
Total Crossovers = 2, Total Mutations = 0
min = 0.180269   max = 0.270883   avg = 0.242145   sum = 1.937161
Global Best Ambiguities Set so far, Iteration 0:
Fitness = 0.270883: 01110101010000001110
-------------------------------------------------------------
Ambiguities set 0 = 807110
Ambiguities set 1 = 246954
Ambiguities set 2 = 132617
Ambiguities set 3 = 98377
Ambiguities set 4 = 1008399
Ambiguities set 5 = 462367
Ambiguities set 6 = 459438
Ambiguities set 7 = 636341
-------------------------------------------------------------
```

Fig. 4 : Initial Iteration of the 3 km Baseline

Ambiguity sets #3 (807367) and #4 (246950) have been totally eliminated by the crossover and mutation processes where the new ambiguity sets produced are 459439 and 462366, respectively. All other ambiguity sets also changed from their original second iteration values, but not as drastically because their fitness values were higher compared to sets #3 and #4. For successive iterations, until iteration 5695, all the changes were very subtle and fitness values increased slowly.

For iteration 5695, all the ambiguity sets had the same value of 452606, with a fitness value of 0.471367. This is one of the criteria needed to stop the iteration, but iteration was continued because the fitness values for succeeding iterations was increasing slowly. For example, iteration 5705 had a fitness value of 0.499698. This is an example of the search reaching a local optimum value. The search continued with every additional iteration showing a greater fitness value until the maximum fitness value of 0.988221 was reached at iteration 7661. The iteration was stopped since the next iteration produced a fitness value greater than 1.0. The final ambiguities set computed included values of 454529 (7 values) and 454528 (1 value). Fig. 5 shows part of the final iteration output. The whole process took 2.71 seconds of CPU time. To reach this final iteration, a total of 15284 crossover and 1107 mutation processes were performed. The total number of these two processes shows the number of changes that are needed to be performed on all eight sets of ambiguities. A higher number of these processes will consume more processing time.

Even though not all 8 sets have the same value, it is apparent that the true ambiguity must be the value 454529 since it occurs 7 times. The total number of possible searchs for this problem is 2^{20}, which is a total of 1, 048, 576 combinations. The PRSS search required 7661 iterations to converge to a global optimum and this is equivalent to 0.73% of the total search space.

```
-------------------------------------------------------------------------------------
Iteration 7661                                              Iteration 7662
num  string                fitness  string                fitness
-------------------------------------------------------------------------------------
1)10000001111101110110  0.970624 | 10000001111101110110  0.970624
2)10000001111101110110  0.970624 | 00100001111101110110  0.988221
3)10000001111101110110  0.970624 | 10000001111101110110  0.970624
4)00000001111101110110  0.935461 | 10000001111101110110  0.970624
5)10000001111101110110  0.970624 | 10000001111101110110  0.970624
6)10000001111101110110  0.970624 | 10000001111101110110  0.970624
7)10000001111101110110  0.970624 | 00000001111101110110  0.935461
8)10000001111101110110  0.970624 | 10000001111101110110  0.970624
-------------------------------------------------------------------------------------
Iteration 7661 Accumulated Statistics:
Total Crossovers = 15284, Total Mutations = 1107
min = 0.970624   max = 0.988221  avg = 0.966229  sum = 1.958845
Global Best Ambiguities Set so far, Iteration 7661:
Fitness = 0.988221: 00100001111101110110
-------------------------------------------------------------------------------------
Ambiguities set 0 = 454529
Ambiguities set 1 = 454529
Ambiguities set 2 = 454529
Ambiguities set 3 = 454528
Ambiguities set 4 = 454529
Ambiguities set 5 = 454529
Ambiguities set 6 = 454529
Ambiguities set 7 = 454529
```

Fig. 5 : Final Iteration of 3 km Baseline for Epoch 1

The same search was performed using the 2nd, 3rd and 4th epoch of measurements. The next epoch search can be used as external criteria to validate the result of the first epoch. This external validation can only be performed if no loss of lock occurs between the consecutive epochs of measurement. If this is the case, then the ambiguity number for consecutive epochs are the same. For Epoch 2, 3 and 4, the algorithm required the same number of iterations crossovers and mutations, with slightly different fitness values, to converge to a global optimum where all eight ambiguities took the value 454530. Tables 4, 5 and 6 show summaries of the 3, 12 and 21-km baselines results, respectively.

Table 4 : Summary of 3 km Baseline Processing

Epoch	Set Ambiguities Solved	Iterations	Time (sec)	%Search Space
1	454529(7),454528(1)	7661	2.712	0.73
2	454530(8)	6364	2.253	0.61
3	454530(8)	6364	2.253	0.61
4	454530(8)	6364	2.253	0.61

Table 5 : Summary of 12 km Baseline Processing

Epoch	Set Ambiguities Solved	Iterations	Time (sec)	% Search Space
1	40021(8)	2464	0.872	3.76
2	40004(6), 7236(2)	2482	0.879	3.79
3	40004(5),40020(3)	2317	0.878	3.54
4	40004(5),40020(3)	2317	0.878	3.54

Table 8 : Summary of 21 km Baseline Processing

Epoch	Set Ambiguities Solved	Iterations	Time (sec)	% Search Space
1	1906368(8)	4331	1.165	0.21
2	1906394(7),1906392(1)	3497	3.363	0.17
3	1906399(6) 1906398(1) 1905886(1)	5845	2.812	0.28
4	1906397(8)	161	0.155	0.01

True to its name, the PRSS algorithm is very random in nature. As explained above, the PRSS algorithm depends on two parameters that are the probabilities of *crossover* (P_c) and *mutation* (P_m). Various combinations were tested for each baseline and no clear-cut combination of P_c and P_m were found. Various combinations were tested for the 21-km baseline. The best range for P_c is between 0.9 and 0.5. P_c less than 0.5 will cause the PRSS algorithm to fail. The probability of P_m has a more random value for various P_c since a different P_c will produce a different P_m range. For example, P_c = 0.9 will result in the best range for P_m between 0.001 and 0.009. But for P_c = 0.8, the value of P_m = 0.001 will cause the search to fail. Instead the best range is from 0.003 to 0.009.

P_{rec}	P_m
0.9	0.001 - 0.009
0.8	0.0003 - 0.009
0.7	0.0007 - 0.009
0.6	0.001 - 0.01
0.5	0.0003 - 0.007

Fig. 6 : P_c and P_m Combination for the 21 km Baseline.

VII. CONCLUSION

It has been shown that a pseudo-randomized search coupled with the AFM, produces a very efficient search of the correct set of ambiguities for baseline lengths ranging from 3 km to 21 km.

The primary conclusion that can be made from this research is that the PRSS algorithm is capable of resolving the carrier phase ambiguity under the three main constraints that had been set up originally. These constraints are that the ambiguities must be resolved under the following conditions :

a. On-The-Fly, that is, no initialization is required
b. One Epoch measurement; this will assure an instantaneous position.
c. L_1 C/A code measurement; this will ensure handheld receiver can be used.

The PRSS algorithm also can be considered as very efficient in performing its ambiguity search. The efficiency is based on the processing time and search space window requirement. For all the test search's, no more than four seconds of processing time was needed to resolve the ambiguity correctly and, for the search space, no more than three percent of the space window are used.

The PRSS algorithm is a very robust search algorithm. The algorithm does not need good starting initial ambiguities to resolve the ambiguities correctly. When different initial ambiguities were used and tested, all the ambiguities for the 3-km and 12-km baseline were resolved correctly. More tests should be performed to determine the effect of good initial starting initial ambiguities on this algorithm. This is true especially under kinematic mode where some form of filtering can be implemented where next position can be estimated.

The PRSS algorithm was designed not solely for single frequency receivers but can be easily adapted to dual frequency receivers. It is expected that the PRSS algorithm will perform better with the dual frequency receivers.

VIII. REFERENCES

Counselman C. and Gourevitch S. (1981). Miniature Interferometer Terminals for Earth Surveying : Ambiguity and Multipath with Global Positioning System. IEEE Transactions on Geoscience and Remote Sensing, Vol. GE-19, No.4, October.

Euler H. and Landau H. (1992). Fast Ambiguity Resolution On-the-fly for Real Time Applications. Proceedings of the Sixth International Geodetic Symposium on Satellite Positioning, Columbus, Ohio, March 17-20.

Frei E. and Beutler G. (1990). Rapid Static Positioning Based on the Fast Ambiguity Resolution Approach: The Alternative to Kinematic Positioning. Proceedings of the 2nd International Symposium on Precise Positioning with GPS, Ottawa, Canada, Sept. 3-7.

Han S. and Rizos C. (1996). Improving the Computational Efficiency of the Ambiguity Algorithm. Journal of Geodesy, Vol. 70.

Hatch R. (1990). Instantaneous Ambiguity Resolution. Proceedings of the IAG International Symposium No. 107 on Kinematic Systems in Geodesy, Surveying and Remote Sensing, Springer Verlag, New York.

Lachapelle G., M.E. Cannon & G. Lu (1992). Ambiguity Resolution on-the-fly - A Comparison of P Code and High Performance C/A Code Receiver Technologies. Proceedings of ION GPS-92, Albuquerque, N.M. 16-18 September.

Landau H. and Euler H. (1992). On-The-Fly Ambiguity Resolution for Precise Differential Positioning. Proceedings of ION GPS-92, Albuquerque, N.M. 16-18 September.

Mader G. (1990). Rapid Static and Kinematic Global Positioning System Solutions Using the Ambiguity Function Technique. Journal of Geophysical Reserach, Vol. 97 N. B3.

Mader G. (1992). Kinematic GPS Phase Initialization using the Ambiguity Function. Proceedings of the Sixth International Geodetic Symposium on Satellite Positioning, Columbus, Ohio, March 17-20.

Remondi B. (1984). Using the Global Positioning System (GPS) Phase Observable for Relative Geodesy : Modeling, Processing and Results. Doctoral Dissertation, Center for Space Research, University of Texas at Austin.

Remondi B. (1991). Pseudo-Kinematic GPS Results using the Ambiguity Function Method. Navigation, Vol. 38, No. 1, Spring.

Teunissen P. (1994). A New Method For Fast Carrier Phase Ambiguity Estimation. Proceedings of IEEE PLANS, Las Vegas, Nevada, April 11-15.