

# Epoch-By-Epoch 방법에 의한 미지정수 해결을 위한 기초연구 Basic Study for Ambiguity Resolution using Epoch-By-Epoch Algorithm

박정현<sup>1)</sup> · 강준목<sup>2)</sup>

Park, Joung Hyun · Kang, Joon Mook

<sup>1)</sup> 동강대학 토지정보관리과 겸임교수(E-mail : [parkjh70@empal.com](mailto:parkjh70@empal.com))

<sup>2)</sup> 충남대학교 공과대학 토목공학과 교수(E-mail : [jmkang@cnu.ac.kr](mailto:jmkang@cnu.ac.kr))

## Abstract

GPS double difference, carrier phase measurements are ambiguous by an unknown inter number of cycle. High precision, relative GPS positioning can be obtained from a short time span of data if the integer double difference ambiguities can be determined efficiently. In this study, we used a ambiguities resolution method indicated by Clyde Goad. And I compared with ambiguities using Lambda method. The difference of ambiguities between two method is below ±1.

## 1. 서 론

기준타원체가 GRS80으로 변경되고 인터넷을 통해 상시관측소 성과를 얻을 수 있게 됨에 따라 GPS 측량의 효율성이 크게 증가되고 있으며, GPS를 이용한 효율적인 측량방법 및 기선해석 프로그램 개발을 위해 많은 노력을 기울이고 있다.

정지측량의 경우 1시간 이상 관측된 데이터를 해석하여 mm의 정밀도를 얻을 수 있지만, 이동측량의 경우는 관측시간이 수 분 이내이기 때문에 단기간에 미지정수를 해결해야 하며, 특히 도심지 등과 같이 위성 시준에 장애물이 많은 지역에서는 이동중에 신호가 자주 단절되기 때문에 미지정수 결정에 많은 어려움이 있다.

본 연구에서는 Clyde Goad에 의해 소개된 Epoch-By-Epoch 방법을 이용하여 미지정수를 해결하고 Lambda 방법에 의해 해결된 실제 미지정수를 기준으로 Epoch별 변동량을 분석하고자 한다.

## 2. 미지정수 결정 이론

GPS 측량 결과를 기선해석하기 위해 대부분 이중차분을 형성하여 위성과 수신기의 시계오차, 전리층 오차, 대류층 오차 등을 소거하거나 줄이게 되며, 반드시 미지정수를 해결해야만 mm의 정확도를 얻을 수 있다. 이주차 GPS 수신기에 대한 관측방정식은 다음과 같이 나타낼 수 있다.

$$\begin{aligned}
P_1 &= \rho^* + I - e_1 \\
\Phi_1 &= \rho^* - I + \lambda_1 N_1 - \epsilon_1 \\
P_2 &= \rho^* + (f_1/f_2)^2 I - e_2 \\
\Phi_2 &= \rho^* - (f_1/f_2)^2 I + \lambda_2 N_1 - \epsilon_2
\end{aligned}
\tag{1}$$

단기선의 경우 전리층의 영향(I)은 거의 동일하기 때문에  $I=0$ 이라고 가정하면 식(2)와 같이 표시된

다.

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & \lambda_1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \rho^* \\ N_1 \\ N_2 \end{bmatrix} = \begin{bmatrix} P_1 \\ \Phi_1 \\ P_2 \\ \Phi_2 \end{bmatrix} - noise \quad (2)$$

그리고, P코드의 오차는 30cm, L1, L2 반송파는 0.5cm라 가정하면 경중률 매트릭스는 다음과 같다.

$$C = \begin{bmatrix} 1/0.3^2 & 0 & 0 & 0 \\ 0 & 1/0.005^2 & 0 & 0 \\ 0 & 0 & 1/0.3^2 & 0 \\ 0 & 0 & 0 & 1/0.005^2 \end{bmatrix} \quad (3)$$

경중률을 고려하여 식(2)를 풀면 실수 미지정수를 얻을 수 있으며, Cycle Goad는 식(4), (5)와 같이 매개변수 K1, K2를 계산하여 실수 미지정수를 계산하였다.

$$K_1 = round(n1 - n2) \quad (4)$$

$$K_2 = round(60 \times n1 - 77 \times n2) \quad (4)$$

$$N_2 = round((60 \times K_1 - K_2)/17)$$

$$N_1 = round(N_2 + K_1) \quad (5)$$

여기서, n1, n2 : L1과 L2 반송파에 대한 실수 미지정수  
 round() : 계산된 값을 정수로 변환하여 값 반환

### 3. 프로그램 개발

수신기마다 독자적인 형식을 사용함으로 본연구에서는 기준점과 이동점에 대한 RINEX 형식의 관측 및 항법데이터를 이용하여 미지정수를 결정하고 기선벡터를 산출하도록 하였으며, 기준점 좌표가 입력되지 않은 경우는 헤더파일에 있는 기준점의 근사좌표를 이용하도록 하였다. 관측방정식은 식(2)와 같이 각 위성별로 이중차를 형성하였으며, Clyde Goad에 의한 미지정수 결정 알고리즘을 이용하여 미지정수를 결정하도록 하였다. 그리고, 결정된 미지정수를 이용하여 3차원 기선벡터를 산출하도록 하였다.

프로그램은 RINEX 형식의 관측 및 항법데이터를 읽어들이는 부분(CRinex), 자료처리에 대한 기본환경을 점검하여 그 관측값의 사용유무를 점검하는 부분(ConFig), 미지정수 해결 부분(CAmbiguity), 해결된 미지정수를 이용하여 기선벡터를 산출하는 부분(CDDFixed)으로 나뉘어져 있다.

각각의 부분은 C++ 클래스로 만들었으며, 데이터의 입출력을 위해 Dialog 기반으로 프로그램을 개발하였고 Visual C++6.0를 이용하여 실행파일을 만들었다.

개발된 프로그램은 기본적으로 자료처리된 결과에서 사이클슬립을 점검할 수있지만 보정하진 않았으며, 단거리 전용으로 전리층과 대류층 오차, 다중경로등에 대한 보정 알고리즘을 첨가되지 않았기 때문에, 사이클 슬립이 발생한 경우나 다중경로의 영향이 심각한 경우에는 문제 발생의 여지가 있다.

### 4. 기선해석 및 결과분석

#### 4.1 미지정수 해결

Clyde Goad에 의해 발표된 one epoch 미지정수 해결 알고리즘을 이용하여 각 epoch별로 미지정수를 해결한 후, 광역(Wide lane) 미지정수를 계산하였으며, Lambda 방법으로 계산된 이중차 미지정수와 비

교한 결과는 그림 1과 같다. 관측시 5대 위성(28,20,7,4,11)이 관측되었지만, 위성고도가 낮은 11번 위성을 제거하였으며, 위성고도가 가장 높은 28번 위성을 기준위성으로 하여 미지정수를 계산하였다.

one epoch 이중차 미지정수의 경우 광역에서 1과장 이상 차이가 나지 않음을 알 수 있었다. 10분동안 관측된 one epoch 이중차 미지정수를 평균하여 광역 미지정수를 계산한 결과는 Lambda 방법으로 계산된 미지정수와 동일하였다. 그러나, 위성고도가 낮은 11번 위성의 경우는  $N1 = 20$ ,  $N2 = 5$ 로 계산되어 실제 미지정수  $trueN1 = 16$ ,  $trueN2 = 2$ 와 큰 차이를 나타내었으며, 광역으로 1과장의 차이를 나타내었다.

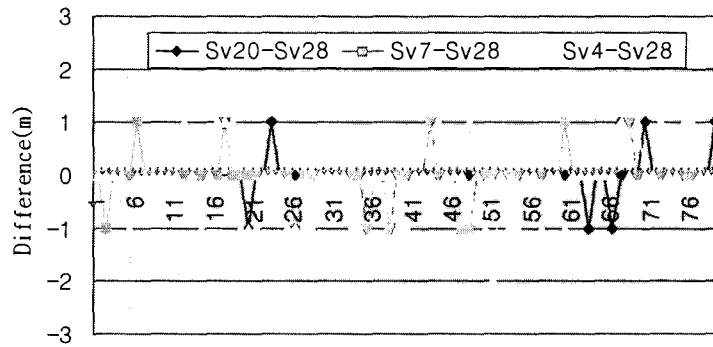


그림 1. One Epoch으로 계산된 미지정수와 Lambda 방법으로 계산된 미지정수와의 광역 미지정수 차이

#### 4.2 기선해석 정확도 분석

그림 2는 one epoch에 계산된  $N1$ ,  $N2$ 미지정수를 이용하여 이중차 정수해를 산출한 결과로, 미지정수가 올바르게 해결되어 10cm 이하로 계산된 경우도 있지만 기선벡터가 1m 이상 변화하고 있으며, 경우에 따라서는 3m까지 차이를 나타내고 있어서 one epoch에 미지정수를 올바르게 해결하는 것이 쉽지 않음을 알 수 있었다.

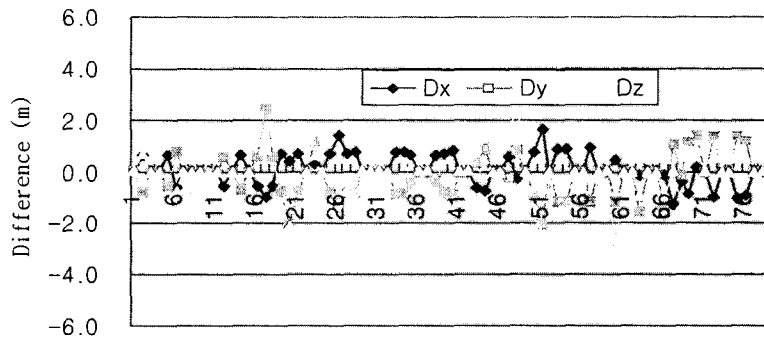


그림 2. One Epoch 이중차 미지정수를 이용한 정수해 계산

10분간 관측된 one epoch 이중차 미지정수를 평균하여 미지정수를 계산하고, 이를 이용하여 각 epoch 별로 3차원 기선벡터를 산출하였으며, Lambda 방법으로 기선해석한 결과와 기선벡터의 차이를 나타낸 결과는 그림 3과 같다. 미지정수가 Lambda 방법과 동일한 결과를 나타내었기 때문에 6cm 이하의 차이를 보이고 있다. 그러나, 위성고도각이 낮은 11번 위성을 포함한 경우는 미지정수가 올바르게 결정되지 않았기 때문에 그림 4와 같이 중심부분에서는 수 cm를 나타내지만, 중심에서 멀어질수록 수평 5cm, 높이 20cm의 상대적으로 큰 오차를 나타내고 있다.

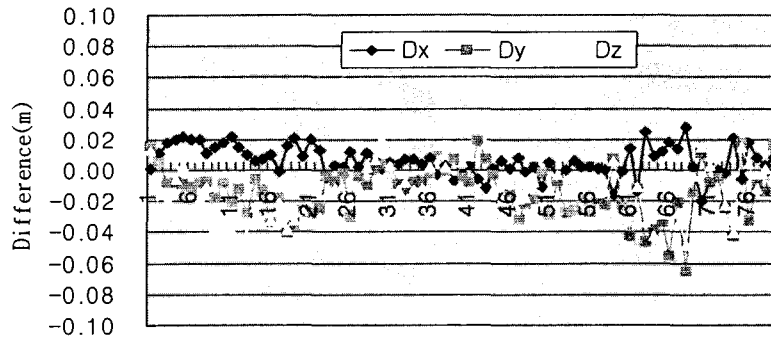


그림 3. 10분간 계산된 one epoch 미지정수를 평균한 후, 각 epoch마다 계산된 기선벡터

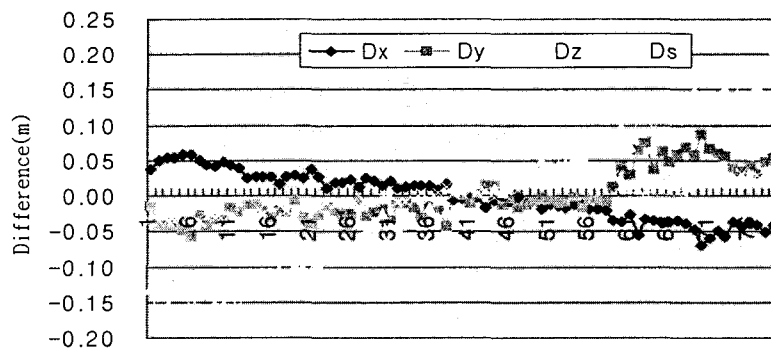


그림 4. 낮은 위성(Sv11)을 포함한 경우에 대한 기선벡터

#### 4. 결과 및 고찰

1. Clyde Goad 에 의한 Epoch-By-Epoch 알고리즘을 이용하여 미지정수를 해결하고 3차원 기선벡터를 산출할 수 있는 프로그램을 개발하였다.
2. Epoch-By-Epoch 방법에 의해 계산된 미지정수는 Lambda 방법에 의해 계산된 미지정수와 비교하여 광역에서 1파장 이상 차이가 나지 않으나, one Epoch에 계산된 미지정수를 이용하여 기선해석하면 1m의 기선오차를 포함할 수 있음을 알 수 있었다.
3. 낮은 위성을 제거하고 10분이상 누적하여 미지정수를 계산하면 실제 미지정수와 거의 동일한 미지정수를 얻을 수 있었으나, 위성고도각이 낮은 경우에는 미지정수가 올바르게 해결되지 않기 때문에 위성고도에 따른 오차범위를 설정하고 검색알고리즘을 이용하여 미지정수를 해결해야 함을 알 수 있었다.

앞으로 Epoch-By-Epoch 방법에 의해 계산된 미지정수의 정확도 향상을 위해서는 보다 상세한 연구가 필요할 것으로 사료된다.