

저가형 측위장치에 RTS 보정정보의 실시간 LSTM 예측 기능 구현을 통한 PPP

Real-time LSTM Prediction of RTS Correction for PPP by a Low-cost Positioning Device

김범수¹ · 김민규¹ · 김정래^{1*} · 부성춘² · 이철수²

¹한국항공대학교 항공우주 및 기계공학부

²LIG넥스원 C4I 연구소

Beomsoo Kim¹ · Mingyu Kim¹ · Jeongrae Kim^{1*} · Sungchun Bu² · Chulsoo Lee²

¹*School of Aerospace and Mechanical Engineering, Korea Aerospace University, Gyeonggi-do 10540, Korea

²C4I Laboratory, LIG Nex1, Gyeonggi-do, 16911, Korea

[요 약]

IGS (international gnss service)에서는 GNSS (global navigation satellite system) 위성의 항법메시지에 적용할 수 있는 RTS (real-time service) 궤도 및 시계 보정정보를 제공한다. 하지만, 인터넷 단절이 발생하면 RTS 값을 수신할 수 없으므로, 안정적인 PPP (precise point positioning)를 수행하기 위해 신호 단절이 발생한 경우 RTS 보정정보를 예측해서 사용해야 한다. 본 논문에서는 실시간으로 신호 단절 구간에서 LSTM (long short-term memory) 알고리즘으로 궤도 및 시계 보정정보를 예측하여 PPP를 진행하였다. 연산 처리 속도가 빠르지 않은 Raspberry Pi (RPI)에 LSTM 알고리즘을 구현하여 예측성능을 분석하였다. 다항식 예측기법과 비교하여 LSTM은 장기간 예측에서 우수한 성능을 보였다.

[Abstract]

The international gnss service (IGS) provides real-time service (RTS) orbit and clock correction applicable to the broadcast ephemeris of GNSS satellites. However, since the RTS correction cannot be received if the Internet connection is lost, the RTS correction should be predicted and used when a signal interruption occurs in order to perform stable precise point positioning (PPP). In this paper, PPP was performed by predicting orbit and clock correction using a long short-term memory (LSTM) algorithm in real-time during the signal loss. The prediction performance was analyzed by implementing the LSTM algorithm in RPI (raspberry pi), the processing speed of which is not high. Compared to the polynomial prediction model, LSTM showed excellent performance in long-term prediction.

Key word : IGS RTS, LSTM, GNSS, GPS, PPP, Satellite navigation.

<https://doi.org/10.12673/jant.2022.26.2.119>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 25 March 2022; Revised 6 April 2022
Accepted (Publication) 19 April 2022 (30 April 2022)

*Corresponding Author: Jeongrae Kim

Tel: +82-2-300-0110

E-mail: jrkim@kau.ac.kr

I. 서 론

GNSS 위성을 사용하여 PPP를 수행하면 사용자의 위치를 높은 정확도로 추정할 수 있다. 하지만 방송궤도력만 사용하여 계산된 GNSS 위성의 궤도와 시계의 정확도는 약 100 cm, 5 ns RMS[1]로 PPP에 적합하지 않다. 따라서 추가 보정을 통해 정밀 궤도와 시계를 구해야 하는데, 2013년부터 IGS에서 인터넷을 통해 제공하는 RTS 보정정보를 사용하여 궤도 및 시계 정보를 실시간으로 보정할 수 있다.

IGS RTS 보정정보를 방송궤도력에 적용하면 PPP를 수행하기에 충분한 5 cm, 300ps RMS[1] 정확도의 실시간 GNSS 위성 궤도 및 시계를 얻을 수 있다. 하지만 인터넷 연결 상태 및 RTS 송신 상태에 의해 RTS 보정정보를 사용할 수 없는 경우 위성 궤도 및 시계 오차가 급증하게 된다. 따라서 안정적으로 PPP를 수행하기 위해서는 RTS 보정정보의 예측 알고리즘이 구현되어야 한다.

이전부터 다양한 방법을 사용한 RTS 보정정보 예측 연구가 수행되었다. 먼저 다항식 모델로 RTS 단기간 예측을 진행하는 연구가 제안되었는데, 시간이 지날수록 예측 오차가 급증하는 경향을 보였다[2]. IGU (igs ultra-rapid) 시계 오프셋을 조화함수로 모델링하여 RTS 시계 보정정보를 예측하는 연구와[3] ARIMA (auto-regressive integrated moving average) 모델을 사용하여 시계 보정정보 예측을 진행한 연구가 있었으며[4], NN (neural network) 또는 GA (genetic algorithm)을 ARMA (auto-regressive moving average)와 결합하여 예측을 진행한 사례도 있었다[5, 6]. 그리고 BiLSTM (bidirectional long short-term memory)을 사용하여 시계 보정정보를 예측한 사례도 존재하였다[7, 8].

본 연구에서는 LSTM 네트워크를 사용하여 RTS 궤도 보정정보를 연산 속도가 빠르지 않은 저가형 측위장치에서 실시간으로 예측하는 연구를 수행하였다. 기존 연구와 달리, 실시간으로 RTS 보정정보를 예측하고 PPP에 사용될 수 있도록 RPI에서 RTS 보정정보 실시간 예측 기능을 구현하였다. 연산량이 많이 요구되는 LSTM 기법을 RPI에서 구현하기 위해, 연산량이 많은 네트워크 훈련은 데스크탑에서 진행하고 훈련된 네트워크를 RPI로 이식하는 방법을 사용하였다. RTS 보정정보 예측 알고리즘은 C 언어 기반의 오픈 소스 위치추정 프로그램 RTKLIB을 수정하여 구현하였다[9].

저가형 수신기와 RPI를 이용하여 실시간 PPP를 수행하는 도중에 인위적으로 신호 단절을 발생시키는 실험을 진행하였다. LSTM 네트워크의 예측성능은 다항식 모델과 비교되었으며, PPP 정확도 분석도 수행되었다.

II. IGS RTS 보정정보

IGS RTS 보정정보는 RTCM (radio technical commission for

maritime services) SSR (state space representation) 형식으로 NTRIP (network transport of RTCM via internet protocol) 클라이언트를 통해 전송된다. 사용자는 인터넷 환경에서 NTRIP 클라이언트를 통해 실시간으로 GNSS 궤도 및 시계 보정정보를 생성 관리 기관에 따라 각각 60초, 10초 또는 5초 간격으로 수신할 수 있다.

RTS 궤도 보정정보는 radial, along-track 그리고 cross-track 방향의 보정정보와 보정정보의 변화율이 포함된다. 따라서 방송궤도력에 ECEF (earth-centered earth-fixed) 좌표계의 변환 후 적용해야 한다. 궤도 보정정보 좌표계의 무게 중심이 CoM (center of mass)인 경우 PCO (phase center offset) 보정도 필요하다.

RTS 보정정보를 방송궤도력에 올바르게 적용하기 위해서는 IOD (issue of date) 비교를 통해 적용 가능 여부를 판단해야 한다. IOD는 RTS 궤도 보정정보에 포함되어 있는데, 방송궤도력과 비교하여 일치하는 방송궤도력에 적용해야 위성의 정밀 궤도 및 시계를 계산할 수 있다.

III. LSTM 알고리즘

RNN (recurrent neural network) 기법에서 장기간 데이터를 다루는 경우 역전파 과정에서 발생하는 기울기 소실 문제에 의해 가중치의 갱신이 제대로 이루어지지 않는다[10]. LSTM은 이러한 RNN 기법의 장기 의존성 문제를 해결하기 위해 도입된 딥러닝 기법이다. 하나의 LSTM cell은 forget, input, 그리고 output의 총 3개의 게이트로 이루어져 있으며, 각각의 게이트와 현재 입력 그리고 이전 hidden state를 통해 cell state를 제어하고 output을 계산한다.

하나의 LSTM cell에서는 식 (1)~(6)을 통해 다음 cell로 전달할 cell state와 hidden state를 계산한다[11].

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{1}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{2}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{3}$$

$$\tilde{C}_t = \tanh(W_C x_t + U_C h_{t-1} + b_C) \tag{4}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{5}$$

$$h_t = o_t * \tanh(C_t) \tag{6}$$

식 (1)~(4)에서 x_t 는 현재의 입력이며 f_t, i_t, o_t 는 각 게이트의 출력, \tilde{C}_t 는 cell state의 추정값, C_t 와 h_t 는 각각 업데이트된 cell state와 hidden state이다. $f_t, i_t, o_t, \tilde{C}_t$ 에 대해 W_f, W_i, W_o, W_C 는 x_t 를, U_f, U_i, U_o, U_C 는 바로 이전 LSTM cell로부터 전달받은 h_{t-1} 을 매핑하는 가중치이다. 그리고 b_f, b_i, b_o, b_C 는 각각의 바이어스이다[11]. 식 (5)는 cell state를 갱신하는 과정으로, forget 게이트

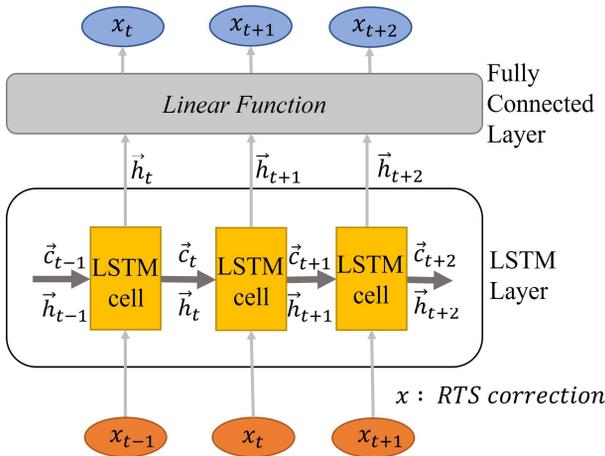


그림 1. LSTM 네트워크 구조
Fig. 1. Architecture of LSTM network

에서 구한 0~1 사이의 출력 f_i 를 C_{i-1} 에 곱하여 이전 LSTM cell로부터 전달받은 cell state에 저장된 데이터의 유지 비율을 결정하고, input 게이트에서 구한 0~1 사이의 출력 i_i 을 \tilde{C}_i 에 곱하여 새로 업데이트될 데이터의 비율을 결정한다. 마지막으로 식 (6)을 통해, 다음 LSTM cell로 전달될 hidden state와 현재 LSTM cell의 출력 h_i 을 계산한다.

그림 1은 1-step 미래의 예측값을 생성하는 LSTM 네트워크를 나타낸 것인데, LSTM 층의 출력을 Fully-connected 층의 가중치와 바이어스로 이루어진 선형 함수에 입력하여 예측 결과를 생성할 수 있다.

IV. 실험 방법 및 결과 분석

4-1 실험 수행 방법

본 연구에서 사용된 RPI는 raspberry-pi 4B 모델로, 사양은 ARM Cortex-A72, 1.5 GHz 4코어 프로세서에 8 GB 메모리이다. RPI에서 연산량이 많은 LSTM 네트워크 예측 기능을 구현하기 위해, LSTM 네트워크의 훈련, 업데이트 및 예측 기능을 분리하였다. 연산량이 가장 많은 LSTM 훈련은 MATLAB 기반으로 데스크탑에서, 네트워크의 업데이트 및 예측 기능은 C 기반의 RTKLIB에 구현하여 RPI에서 수행되었다[10]. RTKLIB은 오픈소스 실시간 위치추정 프로그램으로 실시간 위성 관측값을 사용하여 위치추정을 진행하는데, 인터넷을 통한 RTS 보정정보의 적용이 가능하다. 실험에 사용된 저가형 수신기는 L1/L2 반송파의 수신이 가능한 이중주파수 기반의 ZED-F9P 모듈이 탑재되어 GPS, GLONASS, Galileo 그리고 BeiDou 수신 기능을 제공한다.

RTS는 60초 간격의 IGS03 궤도 보정정보를 사용했다. LSTM 네트워크는 그림 1과 같이 1-step 뒤의 값, 즉 60초 뒤의 값을 예측하도록 훈련되었다. RPI로 훈련 네트워크의 이식은 LSTM 층과 fully-connected 층의 가중치와 바이어스를 텍스트 파일에 저장하는 것으로 이루어졌다. 주기적으로 데스크탑에서 새로운 보정정보로 훈련된 네트워크를 인터넷을 통해 RPI로 옮기는 것으로 LSTM 네트워크의 업데이트가 가능하다.

또한, 비교 연구를 위해 RPI에 다항식 모델을 사용한 RTS 보정정보의 예측 기능도 구현되었다.

그림 2는 RPI에서 RTKLIB이 구동되어 PPP가 실시간으로

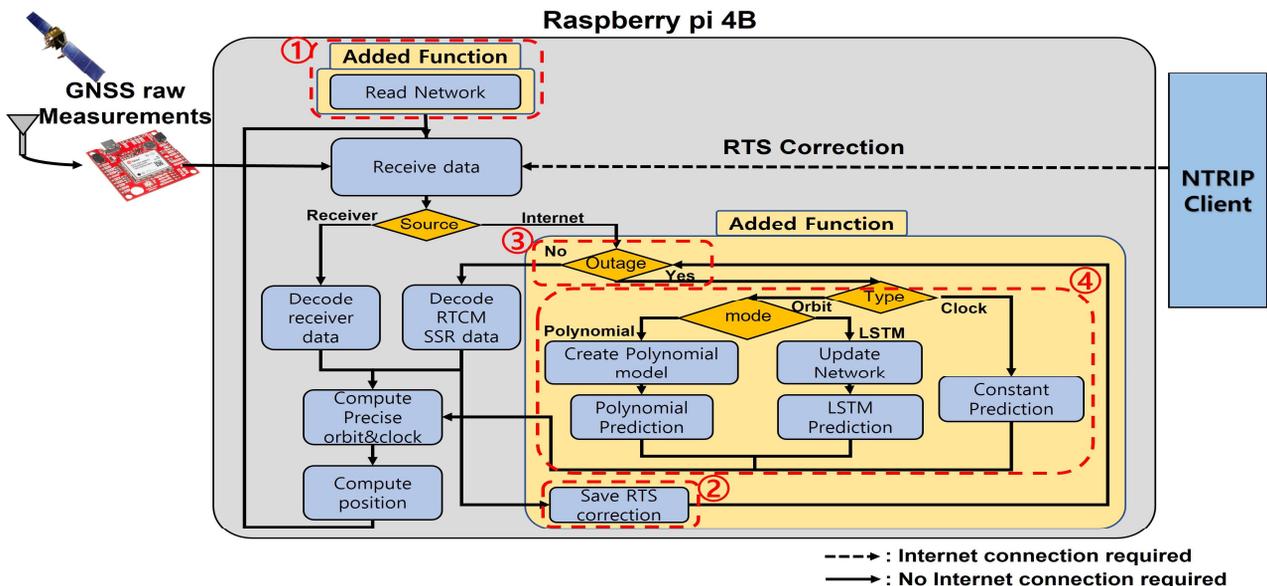


그림 2. RTKLIB PPP의 데이터 흐름 및 추가된 기능
Fig. 2. Data flowchart and added function to RTKLIB PPP

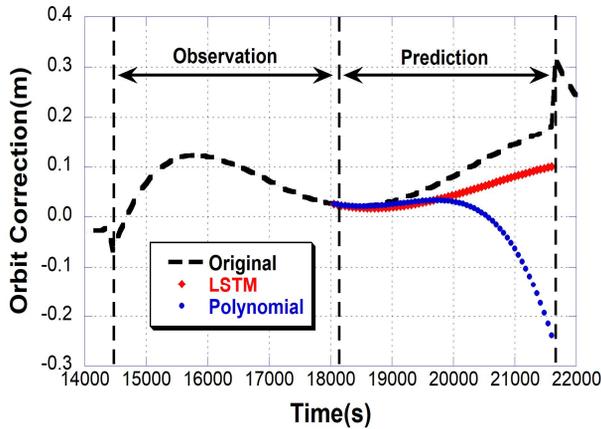


그림 3. PRN 5 RTS 궤도(radial) 관측값 및 예측값
 Fig. 3. Observations and predictions of PRN 5 RTS orbit(radial)

수행될 때 데이터의 흐름을 나타낸 것이다. 수신기와 NTRIP 클라이언트를 통해 RTS 데이터를 수신하고 위성의 실시간 정밀 궤도 및 시계를 계산하여 PPP에 적용한다.

LSTM 네트워크의 업데이트 및 예측 기능을 구현하기 위해 RTKLIB에 몇 가지 기능이 추가되었다[10]. 첫 번째는 훈련 네트워크를 불러들여서 PRN (pseudorandom noise) 별로 저장하는 기능이다 (그림 2 ㉓). 두 번째로는 인터넷을 통해 실시간으로 수신하는 RTS를 저장하는 기능이다 (그림 2 ㉔). 세 번째로는 신호 단절을 감지하는 기능으로, 신호 단절이 발생하면 플래그를 생성하여 신호 단절을 여부를 알린다 (그림 2 ㉕).

마지막으로 RTS를 예측하는 기능인데, 신호 단절이 감지되는 순간 저장된 RTS를 사용하여 예측 모델을 생성하고 예측을 수행한다 (그림 2 ㉖). 예측 결과는 실시간으로 손실된 RTS 데이터를 대신하여 위성 정밀 궤도 및 시계 계산에 사용된다.

또한, 네 가지 기능 외에 RTKLIB의 옵션을 추가하여 예측 방법을 선택할 수 있도록 하였다. LSTM의 경우 저장된 RTS 보정정보를 사용하여 cell state와 hidden state의 업데이트를 진행하고, 다항식 모델의 경우 옵션에 지정된 차수에 따라 다항식 모델을 생성한다.

실시간 실험에서는 GPS 위성만 사용하여 PPP가 수행되었으며, 그림 3과 같이 신호 단절 이전 1시간 관측값으로 LSTM 네트워크의 업데이트를 수행하고 신호 단절 발생 후 1시간 동안 예측이 진행되었다.

본 연구에서는 4차 다항식을 사용하여 다항식 예측 모델을 생성하였는데, 이는 4차의 오차가 가장 작았기 때문이다[12]. RTS 시계 정보는 마지막 값을 예측값으로 사용하는 상수 모델을 사용했다. 그 이유는 다항식 모델을 사용해도 성능에 개선이 없었으며[2], LSTM을 사용하는 경우 연산량의 차이에 비해 성능에 큰 차이가 없기 때문이다.

4-2 RTS 예측 성능 분석

표 1. 궤도 예측 3D 오차 통계 (2022년 3월 1-6일, 단위: meters)
 Table 1. 3D error statistics of orbit predictions (March 1-6, 2022, unit: meters)

	Mar. 1	Mar. 2	Mar. 3	Mar. 4	Mar. 5	Mar. 6	Avg.
LSTM Mean	0.309	0.304	0.348	0.349	0.345	0.325	0.330
Poly. Mean	1.193	1.453	1.978	1.240	3.077	7.754	2.783
LSTM STD	0.083	0.089	0.105	0.094	0.080	0.076	0.088
Poly. STD	0.566	0.642	0.821	0.406	10.217	37.105	8.293

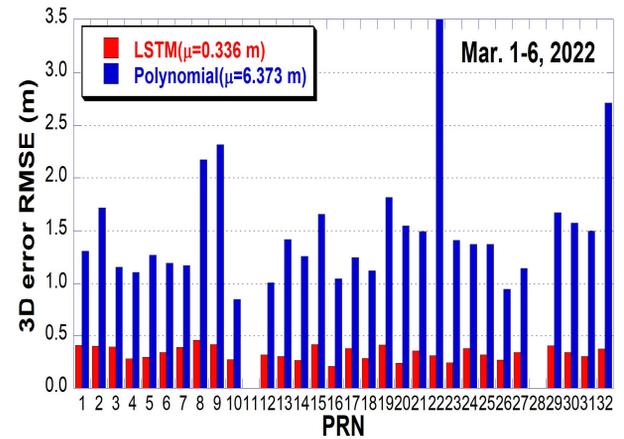


그림 4. RTS 궤도 보정정보 3D 예측 RMSE
 Fig. 4. RTS orbit correction 3D estimation RMSE

LSTM 네트워크 훈련에 사용된 IGS03 RTS 데이터는 2022년 1월 28일부터 2월 3일까지이다. 실험은 2022년 3월 1일부터 6일까지 수행되었으며, 하루에 5번씩 총 30번의 신호 단절을 발생시키고 RTS 예측 오차를 분석하였다.

RTS 보정정보가 없는 PRN 11, 28 위성을 제외하고 총 30개의 위성에 대해서 예측이 진행되었다. 한 시점에서 LSTM 네트워크로 총 30개 위성의 RTS 궤도 보정정보를 예측하는데 RPI에서 평균 0.026초가 소요되었다. 이는 프로그램 내에서 RTS 궤도 보정정보의 해독에 필요한 0.0001초에 비해 상대적으로 크지만, 1초 간격의 PPP에 적용하기엔 충분한 성능이다.

표 1은 일자별로 나눈 RTS 궤도 3D RMS 오차의 평균이다. 모든 결과에서 LSTM의 예측 결과가 우수한 것을 알 수 있으며, 다항식 모델의 3D 궤도 오차는 3월 6일에 7.754 m로 가장 크며, 표준 편차는 37.105 m로 매우 크게 발산한 것을 알 수 있다. 해당 일자에 PRN 22 위성의 관측 데이터 개수가 부족하여 예측값이 발산한 것이 그 원인인데, 이 경우에도 LSTM을 사용하면 기존의 예측 정확도를 유지할 수 있다. 그림 4는 각 PRN에 대해 모든 데이터의 3D RMS 오차의 평균을 나타낸 그래프이다. 총 30번 수행된 1시간 RTS 궤도 예측 결과에서 다항식 모델 대비 LSTM을 사용하는 경우 3D RMS 오차를 평균 94.7% 감소시킬 수 있다.

표 2. PPP 3D RMS 오차 통계 (2022년 3월 1-6일, 단위: meters)
Table 2. 3D RMS error statistics of PPP (March 1-6, 2022, unit: meters)

	Mar. 1	Mar. 2	Mar. 3	Mar. 4	Mar. 5	Mar. 6	Avg.
LSTM Mean	0.159	0.126	0.243	0.341	0.279	0.103	0.209
Poly. Mean	0.868	0.535	1.033	1.728	1.007	1.108	1.047
LSTM STD	0.122	0.111	0.206	0.270	0.274	0.066	0.175
Poly. STD	0.805	0.404	0.949	1.722	0.877	0.888	0.941

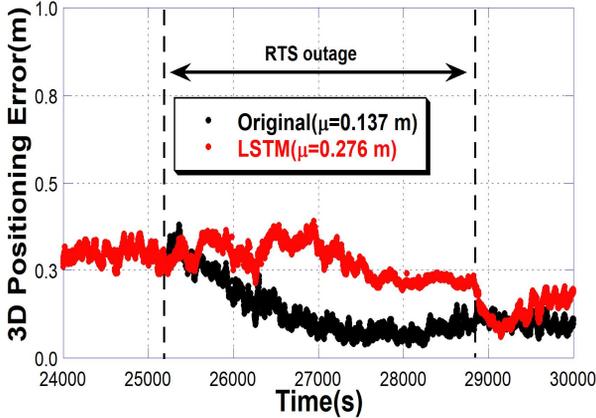


그림 5. 실시간 PPP 3D 오차 (2022년 3월 21일)
Fig. 5. Real-time PPP 3D error (March 21, 2022)

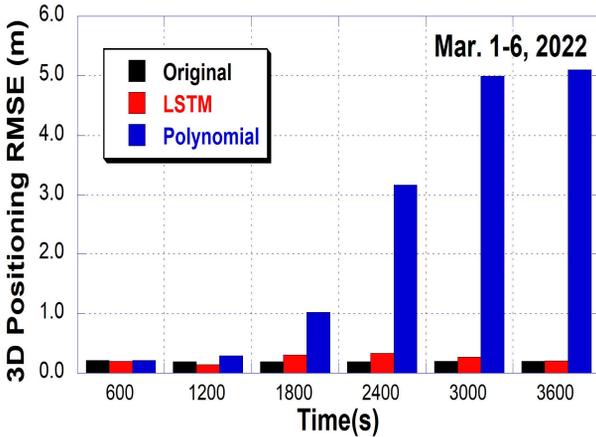


그림 6. 예측 구간별 PPP 3D 평균 오차
Fig. 6. Average PPP 3D error with prediction intervals

4-3 PPP 성능 분석

신호 단절 구간에서 LSTM을 통해 실시간으로 예측된 RTS 보정정보는 방송궤도력에 적용되어 수신기 관측값과 함께 PPP에 사용된다. 그림 5는 1시간의 예측 구간에 대해 LSTM 예측을 진행한 경우와 신호 단절이 발생하지 않은 경우의 실시간 위치 추정 결과이다. LSTM을 사용한 결과 1시간 동안 신호 단절이

전과 같이 0.3 m 수준의 위치추정 정확도를 유지하는 것을 알 수 있다.

그림 6은 10분 단위로 예측 구간을 나누어 3D PPP RMS 오차를 나타낸 그래프이다. 약 10분간의 단기간 예측에서 다항식 모델과 LSTM은 비슷한 성능을 보인다. 하지만 약 20분부터 다항식 모델의 오차가 급증하여 50-60분 구간에서는 LSTM을 사용하면 오차를 95.9% 감소시킬 수 있다.

표 2는 전체 예측 구간에서 LSTM과 다항식 모델의 3D PPP RMS 오차의 일자별 평균을 나타낸 것으로, LSTM 모델은 대부분 평균 0.3 m 이내의 오차를 보이지만, 다항식 모델을 사용하는 경우는 평균 1 m를 초과한다. 전체적으로 LSTM 모델을 사용하는 경우 다항식 모델 대비 약 81.1%의 오차를 감소시킬 수 있었다.

V. 결론

Raspberry Pi (RPI)와 저가형 GPS 수신기를 이용한 실시간 PPP 도중에 임의로 신호 단절을 발생시켜 LSTM 네트워크로 손실된 RTS 보정정보를 예측하고 PPP에 적용하여 위치추정 성능의 분석도 진행하였다. LSTM 네트워크를 사용한 예측에는 많은 연산량이 필요하므로 네트워크 훈련, 업데이트 및 예측 과정을 분리하였다. 데스크탑에서 훈련을 진행하고 RPI로 이식하여 네트워크 업데이트 및 예측을 진행하였다. RTS 보정 정보 예측 기능은 C 기반의 실시간 위치추정 프로그램인 RTKLIB에 구현되었다[10].

1시간의 신호 단절 구간에서 LSTM의 RTS 궤도 예측성능을 다항식 모델과 비교하였으며, 상수 모델을 사용하여 RTS 시계를 예측했다. 다항식 모델 대비 LSTM을 사용한 결과 RTS 3D 예측 오차를 94.7% 오차를 감소시킬 수 있었다. 또한, 실시간 예측 결과를 방송궤도력에 적용하여 실시간 PPP 성능을 비교한 결과 전체 구간에서 81.1%의 오차를 감소시킬 수 있었다. 약 10분까지는 다항식 모델과 LSTM의 성능이 비슷하지만, 예측이 구간 길어질수록 다항식 모델은 급격히 발산하고 LSTM은 오차가 일정 수준을 유지할 수 있었다.

Acknowledgments

본 연구는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원 및 LIG NEX1의 산학협력과제 지원으로 수행되었습니다(NRF-2019R1F1A1062605).

References

[1] International GNSS Service. IGS Real-Time Service Fact

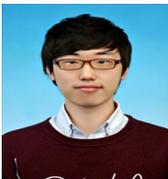
Sheet [Internet]. Available;
<https://kb.igs.org/hc/en-us/articles/201087803-IGS-Real-Time-Service-Fact-Sheet>

- [2] T. Hadas and J. Bosy, “IGS RTS precise orbits and clocks verification and quality degradation over time”, *GPS Solutions*, Vol. 19, No. 1, pp. 93-105, Jan. 2015.
- [3] Z. Nie, Y. Gao, Z. Wang, S. Ji, and H. Yang, “An approach to GPS clock prediction for real-time PPP during outages of RTS stream”, *GPS Solutions*, Vol. 22, No. 14, Nov. 2017. (Online Publication)
- [4] H. Ge, B. Li, T. Wu and S. Jiang, “Prediction models of GNSS satellite clock errors: Evaluation and application in PPP”, *Advances in Space Research*, Vol. 68, No. 1, pp. 2470-2487, Sep. 2021.
- [5] M. G. Kim and J. R. Kim, “Predicting IGS RTS corrections using ARMA neural networks”, *Mathematical Problems in Engineering*, Vol. 2015, No. 851761, pp. 1-11, Jun. 2015.
- [6] M. G. Kim and J. R. Kim. “GA-ARMA model for predicting IGS RTS corrections”, *International Journal of Aerospace Engineering*, Vol. 2017, No. 6316590, pp. 1-7, Mar. 2017.
- [7] M. G. Kim and J. R. Kim, “A short-term prediction method of the IGS RTS clock correction by using LSTM network”, *Journal of Positioning, Navigation, and Timing*, Vol. 8, No. 4, pp. 209-214, Nov. 2019.
- [8] J. H. Kim, M. G. Kim and J. R. Kim, “Prediction Performance Analysis for the IGS RTS Clock Correction using LSTM Network”, in *KSAS 2020 Spring Conference*, Goseong-gun, pp 683-684, Jul. 2020.
- [9] T. Takasu. RTKLIB: An Open Source Program Package for GNSS Positioning [Internet]. Available;
<http://www.rtklib.com/>
- [10] J. H. Kim, Development of real-time precise satellite navigation device using RTS correction with bidirectional LSTM algorithm, M.S. dissertation, Korea Aerospace University, Goyang-si, Feb. 2021.
- [11] Z. Cui, R. Ke, Z. Pu and Y. Wang, “Deep Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction”, in *International Workshop on Urban Computing*, London, pp 1-11, Aug. 2018.
- [12] T. Hadas, J. Kaplon, K. Wilgan, J. Bosy and J. Sierny, “Verification of real-time IGS products and their influence on Precise Point Positioning”, in *GNSS PPP Workshop: Reaching Full Potential*, Ottawa, pp. 1-25, 2013.



김 범 수 (Beomsoo Kim)

2019년 2월: 한국항공대학교 항공우주공학과 (공학사)
 2021년 2월 ~ 현재: 한국항공대학교 항공우주 및 기계공학과 대학원 석사과정
 ※관심분야: 위성항법, 위성궤도



김 민 규 (Mingyu Kim)

2013년 2월: 한국항공대학교 항공우주공학과 (공학사)
 2022년 2월: 한국항공대학교 항공우주공학과(공학박사)
 2022년 3월 ~ 현재: 한국항공대학교 항공우주산업기술연구소 박사후연구원
 ※관심분야: 위성항법, 위성궤도



김 정 래 (Jeongrae Kim)

1991년 2월: 서울대학교 항공우주공학과 (공학사) 2000년 ~ 2002년: 미국 Center for Space Research 연구원
 1993년 2월: 서울대학교 항공우주공학과 (공학석사) 2002년 ~ 2004년: 한국항공우주원 선임연구원
 2000년 5월: 미국 University of Texas at Austin (공학박사) 2004년 ~ 현재: 한국항공대학교 항공우주 및 기계공학부 교수
 ※관심분야: GNSS, 위성항법, 위성궤도



부 성 춘 (Sungchun Bu)

2002년 2월 : 건국대학교 전자정보통신공학과 (공학사) 2007년 4월 ~ 2011년 3월 : 코아로직 선임 연구원
 2004년 8월 : 건국대학교 전자정보통신공학과 (공학석사) 2011년 4월 ~ 현재 : LIG넥스원 수석연구원
 2005년 8월 ~ 207년 3월 : 토필드 연구원
 ※관심분야: 광역보강시스템, 위성항법수신기



이 철 수 (Chulsoo Lee)

2000년 2월 : 한양대학교 전자공학과 (공학사) 2010년 8월 : 충남대학교 전자전파정보통신공학과 (공학박사)
 2002년 8월 : KAIST 전기 및 전자공학과 (공학석사) 2010년 9월 ~ 현재 : LIG넥스원, 수석연구원
 2002년 ~2007년 : ETRI, 연구원
 ※관심분야: 통신 시스템 및 신호처리, 위성항법, 보정항법