



위성 탑재용 천문력 생성 프로그램 개발

이광현¹, 조동현², 김해동³

Development of Planetary Ephemeris Generation Program for Satellite

Kwang-Hyun Lee¹, Dong-Hyun Cho² and Hae-Dong Kim³

University of Science and Technology^{1,3}, Korea Aerospace Research Institute^{2,3}

ABSTRACT

The satellites in orbit use a sun reference vector from solar model based the ephemeris. To get the ephemeris, we use DE-Series, an ephemeris developed by the Jet Propulsion Laboratory (JPL), or the reference vector generation formula proposed by Vallado. The DE-Series provides the numerical coefficients of Chebyshev polynomials, which have the advantage of high precision, but there is a computational burden on the satellite. The Vallado's method has low accuracy, although the sun vector can be easily obtained through the sun vector generation equation. In this paper, we have developed a program to provide the Chebyshev polynomial coefficients to obtain the sun position coordinates in the inertial coordinate system. The proposed method can improve the accuracy compared to the conventional method and can be used for high - performance, high - precision nano satellite missions.

초 록

궤도상에 있는 인공위성은 천문력 기반 태양 모델을 사용하여 기준 벡터를 형성한다. 이를 위해 제트 추진 연구소(JPL)에서 개발한 천문력인 DE-Series, 또는 Vallado가 제안한 기준 벡터 생성식을 사용한다. DE-Series는 체비셰프 다항식의 수치 계수를 제공하는데 정밀도가 높다는 장점이 있지만 인공위성의 탑재 컴퓨터의 계산 부담이 있으며, Vallado 방식은 생성식을 통해 태양 벡터를 간단히 구할 수 있지만 낮은 정밀도를 제공한다. 본 논문에서는 DE-Series를 통해 얻은 태양의 위치를 체비셰프 다항식으로 Curve fitting하여, 관성좌표계에서의 태양 위치좌표를 구할 수 있는 체비셰프 다항식 계수를 제공하는 프로그램을 개발하였다. 기존 방식에 비해 정밀도를 향상시킬 수 있었으며, 제안된 방법은 고성능, 고정밀 초소형위성 임무에 활용될 수 있다.

Key Words : Planetary Ephemeris(천문력), Sun Vector(태양 벡터), Chebyshev polynomials(체비셰프 다항식), Curve fitting(곡선 적합)

† Received : November 6, 2018 Revised : January 28, 2019 Accepted : January 29, 2019

¹ Graduate Student, ² Senior Researcher, ³ Principal Researcher/Professor

³ Corresponding author, E-mail : haedkim@kari.re.kr

I. 서 론

최근 인공위성은 과학의 발전에 따라 과학관측이나 지상 정보수집 등 많은 연구 분야에서 사용되고 있다. 인공위성이 지향성 안테나를 통한 지상국 통신이 필요하거나, 인공위성의 정확한 자세정보를 요구하는 과학임무 수행할 경우 인공위성은 자신의 자세를 정확하게 알 필요가 있다. 인공위성은 본체에 탑재된 자세 센서로부터 얻은 데이터를 바탕으로 스스로 우주 공간에서의 자세 정보를 파악하는 자세 결정을 수행한다. 이 때 사용되는 센서로는 태양 센서(Sun Sensor), 자장 센서(Magnetometer), 지구 센서(Earth Sensor), 별 센서(Star Sensor) 등이 있다.

그 중 태양 센서는 인공위성에서 가장 보편적으로 사용되는 센서로써 전력 소모가 적고 간단한 시스템으로 이루어져 있어 센서 설계와 자세 결정이 간단하다. 이러한 자세 결정 센서는 인공위성의 자세를 변경할 때 자세 정보를 제공하는데 이 경우 센서의 측정으로부터 얻는 측정 벡터와 비교할 기준 벡터가 필요하다. 태양 센서를 사용하여 위성의 자세를 결정할 때, 태양의 위치를 정의하기 위해서 여러 천문력에서 제시하는 태양 벡터를 사용한다. 대표적으로 JPL(Jet Propulsion Laboratory)에서 제공하는 DE-Series와 Vallado가 제시하는 천문력 계산식이 있다.

DE-Series는 DE-421, 430, 432t와 같이 여러 Series로 되어있으며 각각은 정밀도를 보장하는 기간, 계수 등의 차이가 있다[1]. Vallado가 제시하는 천문력 계산식은 DE-Series에 비해 적은 계수를 가지고 태양 벡터를 구할 수 있지만 정밀도가 더 낮다는 단점이 있다[2]. 이 밖에도 태양 벡터를 구하는 방식에는 VSOP(French: Variations Séculaires des Orbites Planétaires)[3], Meeus의 태양벡터 모델[4] 등 다양한 방식이 존재한다.

특히 Vallado가 제시하는 태양 위치 계산식은 정밀도가 낮고, 계산식을 이해하기 위해서는 천문학적 배경지식과, 관련된 수식들을 알아야 한다. 반면 DE-Series와 같은 정밀한 천문력의 경우, 행성의 위치나 속도를 나타내기 위해 테이블 형태의 방대한 데이터를 가진다. 초소형위성의 OBC(On-board computer)에서 많은 데이터를 처리하기 어렵기 때문에 복잡한 연산식을 사용할 수 없다. 따라서 충분한 정밀도를 가지면서 빠른 연산이 가능한 태양 위치 계산식이 요구된다.

본 논문에서는 기존의 문제를 해결하기 위해 천문력 모델로 구한 태양 벡터를 참값으로 Curve fitting을 수행하여 적은 계수로도 태양 벡터를 구할 수 있게 하였다. 그리고 임무기간 전체에 대한 다항식의 계수를 제공하는 프로그램을 개발하였다.

본 논문의 2장에서는 태양 센서, 기존의 천문력 그리고 자세결정 방식을 간단하게 설명하고, 3장에서는

저자들이 제안하는 새로운 천문력 생성기법, 기존 천문력과의 비교 결과 그리고 제안하는 천문력을 생성해주는 프로그램에 대해 설명한다.

II. 태양 벡터

2.1 Sun Sensor

태양 센서는 인공위성의 자세 결정에 보편적으로 사용되고 임의의 자세에서 정해진 자세로의 수정이 필요할 때 기준이 되는 데이터를 제공한다[5]. 태양 센서의 종류는 출력이 디지털인지 아날로그인지에 따라 디지털 태양 센서, 아날로그 태양 센서로 분류되며, 측정치가 얼마나 정밀한지에 따라 고정밀 태양 센서와 저정밀 태양 센서로 분류된다. 일반적으로 아날로그 태양 센서는 저정밀이고, 디지털 태양 센서는 고정밀이다. 정밀도 측면에서는 고정밀이라 하더라도 별센서보다 낮지만 소형으로 제작이 가능하고 전력 소모가 적고 별 센서보다 넓은 FOV(Field Of View)를 갖는다는 장점이 있다. Table 1에서 볼 수 있듯이 고정밀 태양 센서의 경우 0.1 degree 미만의 정밀도를 가지며 저정밀 태양 센서는 0.1-5 degree의 정밀도를 갖는다. 하지만 반대로 고정밀 태양센서는 저정밀 태양 센서보다 좁은 FOV를 가진다[7].

이러한 태양 센서는 앞서 설명하였듯이 슬릿을 통해 들어오는 빛이 이미지 센서에 맺히는 이미지를 이용하여 입사각을 계산하는데 슬릿을 통해 들어오기 때문에 회절과 센서에 맺히는 상을 고려해야 한다. 또한 구조상에 문제가 있을 경우 정밀도에 영향을 주기 때문에 하우징, 이미지 중심 알고리즘 등에 대한 개발도 진행되어 왔다.

태양 센서의 구성은 Fig. 1과 같다. 원형 구멍 또는 슬릿, Complementary metal-oxide-semiconductor Image Sensor(CIS)로 구성되어 있으며, 구동 방식은 Pin hole을 통과하여 들어온 광원이 CIS의 화소에 맺히고 태양광원과 회절광원이 디지털 신호로 변환되는 방식이다. 이렇게 변환된 디지털 신호는 FPGA에서 신호처리 후 계산알고리즘을 통해 입사각을 계산하여 OBC에 제공된다.

Table 1. Sun sensor properties

Sun Sensor	Coarse	Fine
Accuracy (deg)	0.1-5	0.005-0.1
Mass(g)	0.01-2	0.01-2
Power(mW)	0-3	0-3

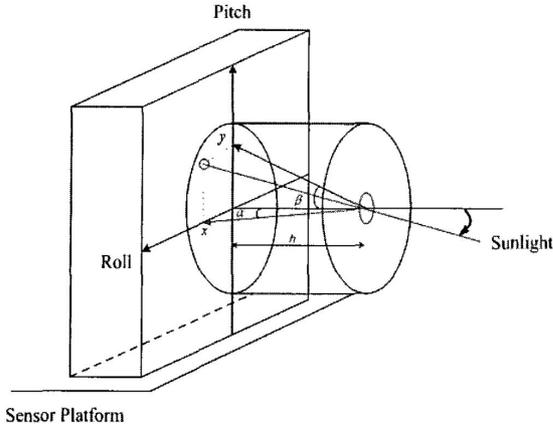


Fig. 1. Configuration of Sun Sensor [6]

광원이 센서면으로 입사되면 태양센서는 식 (1), (2)를 통해 피치 각(α), 롤 각(β)을 구할 수 있다. 이렇게 구한 α, β 를 통해 태양센서의 센서 좌표계에서의 시선 단위 벡터(p^m)은 식 (3)과 같다. 태양센서의 센서 좌표계에서 위성의 동체 좌표계로의 변환행렬과 동체 좌표계에서 지구중심 관성 좌표계로의 변환행렬을 사용하면 지구중심 관성 좌표계에서의 시선 단위 벡터를 구할 수 있다[6].

$$\alpha = \tan^{-1} \frac{x}{h} \quad (1)$$

$$\beta = \tan^{-1} \frac{y}{h} \quad (2)$$

$$p^m = \begin{bmatrix} \sin\alpha\cos\beta \\ \sin\beta \\ \cos\alpha\cos\beta \end{bmatrix} \quad (3)$$

2.2 Attitude Determination Using Sun Sensor

태양 센서는 일반적으로 각속도 정보를 필요로 하지 않는 벡터기반 자세 결정 알고리즘을 사용하여 자세 결정을 수행한다. 벡터기반 자세결정의 대표적인 알고리즘은 TRI-Attitude Determination(TRIAD), Quaternion Estimation(QUEST) 알고리즘이 있다. TRIAD 알고리즘은 평행하지 않은 두 개의 기준 단위 벡터와 관측 단위 벡터가 각각 주어졌을 때 두 벡터 사이의 방향 코사인 행렬을 구하여 자세를 결정하는 방식이다. 간단하며 계산량이 적다는 장점이 있지만 단 두 개의 벡터만을 사용하여 계산할 수 있고 그 두 벡터가 평행할 경우 계산이 불가능하다. QUEST 알고리즘은 최적의 해를 찾을 수 있는 알고리즘이지만 TRIAD 알고리즘에 비해 계산량이 많은 단점이 있다[8].

이러한 태양 센서의 자세 결정 알고리즘을 사용하기 위해서는 Fig. 2와 같은 과정이 필요하다. DE-Series나 VSOP와 같은 모델을 통해 International

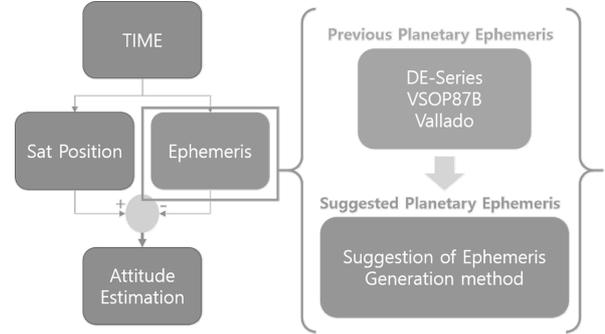


Fig. 2. Block diagram

Celestial Reference Frame(ICRF) 좌표계에서의 실제에 가까운 태양 기준 벡터와 Global Positioning System (GPS)나 Simplified General Perturbations 4(SGP4)등을 사용한 위성의 위치벡터의 차이를 계산하여 위성과 태양 사이의 벡터를 얻는다. 위성으로부터의 태양 벡터와 더불어 자장 센서를 통해 얻은 벡터를 TRIAD, QUEST 알고리즘에 사용하여 위성의 자세를 결정한다.

III. Ephemeris generation

3.1 Previous Planetary Ephemeris

태양, 달, 행성, 항성과 같은 천체들의 시간에 따른 위치, 거리, 광도를 담고 있는 천문력은 주로 천문학에서 사용되는데, 지상에서와 달리 우주에서는 기준 점으로 삼을 물체가 적기 때문에 이러한 정보들이 중요하다. 우주비행체의 항법이나 천문학에 사용되는 천체력의 대표적인 예로는 NASA의 제트추진 연구소(JPL)에서 개발한 'DE-Series' 천문력이 있다. DE-Series 뿐만 아니라 Vallado의 저서에서 제공하는 천문력, 프랑스에서 개발한 VSOP 등 다양한 천문력이 존재한다.

DE-Series는 DE-421, 430, 432t와 같이 여러 Series로 되어있으며 각각은 정밀도를 보장하는 기간, 계수 등의 차이가 있다. 이 모델은 태양, 8개의 주요 행성, 명왕성, 달의 좌표를 포함하고 있으며, 데이터를 체비셰프(Chebyshev) 다항식에 대한 수치 계수 파일로 배포한다. 태양 벡터의 경우 실제값에 가까운 정밀도를 제공하지만 1년의 정보를 얻기 위해서는 많은 데이터를 필요로 한다. 태양 벡터를 구하기 위해서는 달의위치도 알아야 하며 이를 위해서는 32일당 총 1020개의 계수가 필요하므로 1년의 미션을 수행한다면 100kbyte 이상이 요구된다.

Vallado의 저서에서 제공하는 태양 벡터의 경우 많은 계수를 사용하지 않아 적은 데이터로 태양 벡터를 구할 수 있다. Vallado의 저서에서는 0.01 degree의 정밀도를 제공한다고 하지만, 실제로 2018년 1월 1일부터 2019년 1월 1일까지에 대해 Vallado's method

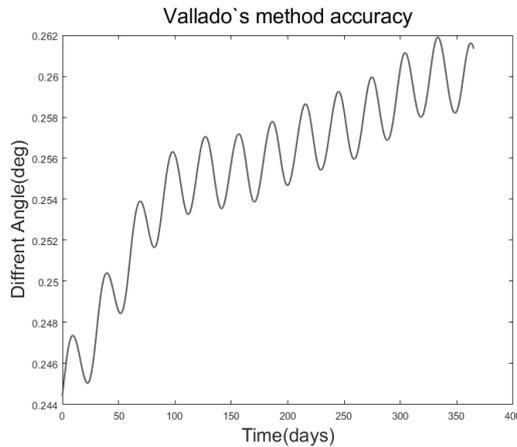


Fig. 3. Vallado's method Accuracy compared to DE-430

를 적용하여 태양 벡터를 구한 결과 DE-Series와 비교했을 경우 0.25 degree의 오차가 있는 것을 Fig. 3을 통해 확인할 수 있었다.

Variations Seculaires des Orbites Planetaries(VSOP)는 프랑스 파리의 Bureau des Longitudes에서 개발된 행성이론으로 시간에 따른 태양계 행성의 장기 변화(secular variation)를 설명하는 모델이다. 개선된 버전이 나오고 있으며 초기 버전인 VSOP82는 Pierre Bretagnon이 1982년에 개발하였고, 단지 궤도요소만을 계산해주었다면 이후 개발된 VSOP87은 2000년 전후 4000년 동안 1"의 정밀도를 보장할 만큼 정밀도가 향상되었을 뿐 아니라 행성의 위치와 궤도요소를 계산하여 제공한다.

Figure 4는 NASA의 SPICE toolkit에 대한 Angular error를 계산한 것으로 VSOP87 버전 중 하나인 VSOP87B와 Vallado의 정밀도를 확인할 수 있다. Table 2에서 볼 수 있듯이 Vallado의 정밀도는 DE-Series를 통해 얻은 결과와 비슷하게 평균 0.23 degree, 최대 0.24 degree로 나타나며 VSOP87B의 경우 5.4×10^{-6} degree의 정밀도를 보여준다. 하지만

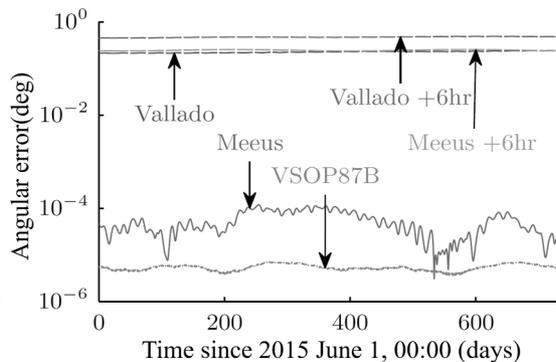


Fig. 4. Sun direction approximation methods and their error relative to SPICE [9]

Table 2. Sun direction approximation methods

Method	Computation time	Angular error	
		Mean [deg]	Max [deg]
Vallado	1	0.23	0.24
VSOP87B	360	5.4×10^{-6}	7.2×10^{-6}

DE-Series와 마찬가지로 계산이 복잡하며 계산시간은 Vallado's method와 비교할 때 360배나 더 오래 걸린다[9].

3.2 Suggestion of Ephemeris generation method

앞에서 제시한 기존의 방식들은 높은 정밀도를 제공하는 천문력의 경우 복잡하거나 많은 데이터를 요구하며, 복잡하지 않은 천문력의 경우 낮은 정밀도를 제공한다. 높은 정밀도를 제공하지만 복잡성은 낮은 천문력을 생성하기 위해 본 논문에서는 JPL의 천문력 모델인 DE-Series에 Curve fitting을 적용하여 압축(Compression)하고 재생(Reconstruction)하여 위성의 임무기간에 대해 높은 정밀도의 태양 벡터를 제공하는 함수를 생성하였다.

3.2.1 Compression

MATLAB® Aerospace Toolbox의 planet-Ephemeris 함수는 JPL의 천문력 모델들에 따라 방향 벡터의 중심과 목표로 하는 행성간의 위치와 속도를 3차원 벡터로 계산해준다. DE-Series는 TDB(Barycentric Dynamical Time) 시간 좌표계를 사용하지만 Planet-Ephemeris 함수에서는 Julian date를 사용하기 때문에 사용자가 UTC(Coordinated Universal Time)를 Julian date으로 변경하여 이를 입력으로 넣으면 타겟 행성의 위치와 속도 벡터를 구할 수 있다. 그리고 km 혹은 Astronomical Unit(AU) 입력에 따라 그에 맞는 단위로 출력한다.

시뮬레이션은 시작시점과 종료시점을 각각 2018년 1월 1일, 2019년 1월 1일로 설정하고 Compression을 수행하였다. Planet-Ephemeris 함수에서 제공하는 태양 벡터를 기준 벡터로 하여 벡터의 크기인 태양과의 거리(R)를 계산하고, 기준벡터를 R 로 나누어 단위 벡터($P_{X,Y,Z}$)로 변환한 다음 태양의 방위각(Azimuth, az)과 고각(Elevation, el)을 식 (4), (5)와 같이 계산한다.

$$az = \tan^{-1}(P_Y/P_X) \quad (4)$$

$$el = \tan^{-1}(P_Z) \quad (5)$$

위 식에서 구한 값들은 시간에 따른 값으로 표현하기 위해 24차의 체비셰프 다항식으로 Curve fitting하여 25개의 계수를 구한다. 체비셰프 Curve

fitting의 경우 전체 시간에 대해 일정한 시간간격의 데이터를 사용하는 것이 아니라 식 (6)과 같이 계산하므로 차수에 따라 fitting에 사용하는 데이터가 다르기 때문에 차수를 얼마로 하여 전체 데이터 셋을 구성했는지에 따라 정밀도에 영향을 준다. N 은 체비셰프 다항식 차수고, I 는 0부터 N 까지의 정수배열이다.

$$Time_{sampling} = \frac{(start + stop)}{2} - \frac{(stop - start) \times \cos\left(\frac{(2 \times I + 1) \times \pi}{2 \times N + 2}\right)}{2} \quad (6)$$

식 (6)으로부터 얻은 $time_{sampling}$ 의 시점에서의 방위각과 고각 데이터를 사용하여 체비셰프 다항식 계수를 계산한다.

3.2.2 Reconstruction

압축을 통해 얻은 계수와 식 (7)을 사용하여 방위각과 고각을 재생(reconstruction)한다. 식 (7)의 N 은 체비셰프 다항식 차수, a_n 은 체비셰프 계수이고 t_{rec} 은 reconstruction time이다.

$$poly = \sum_{n=0}^N a_n \cos(ncos^{-1}(t_{rec})) \quad (7)$$

재구성한 데이터와 DE-430에서 얻은 데이터의 차이는 Figs. 5, 6, 7과 같다. 방위각, 고각, 거리에 대한 평균 오차는 각각 0.00118 degree, 0.00031 degree, 2.001×10^{-5} AU로 압축, 재생 후의 계산 결과가 정밀하다는 것을 알 수 있다.

식 (8)과 같이 Curve fitting한 방위각과 고각을 사용하여 태양 벡터를 구한다.

$$Sun = [\cos(el)\cos(az), \cos(el)\sin(az), \sin(el)] \quad (8)$$

위의 식으로 계산한 태양 벡터와 MATLAB®의 planetEphemeris 함수로 구한 태양 벡터의 사이각(오차)을 식 (9)와 같이 계산한다.

$$\theta = \cos^{-1}(Sun \times P) \quad (9)$$

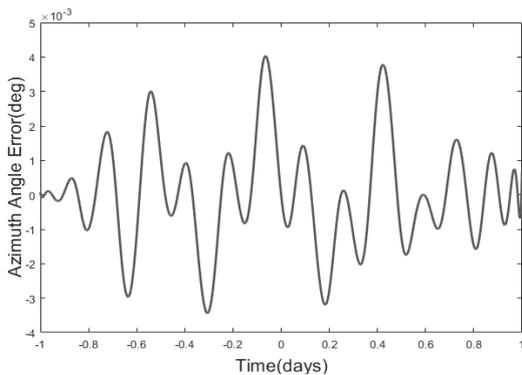


Fig. 5. Reconstruction error of azimuth

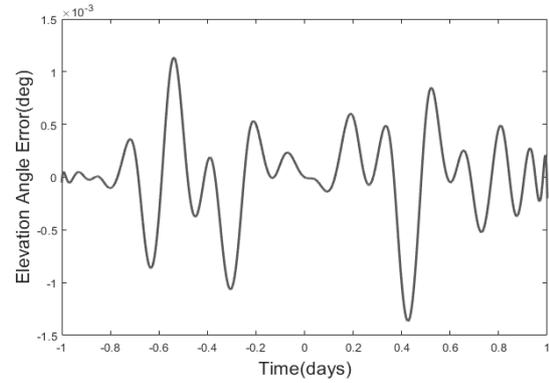


Fig. 6. Reconstruction error of elevation

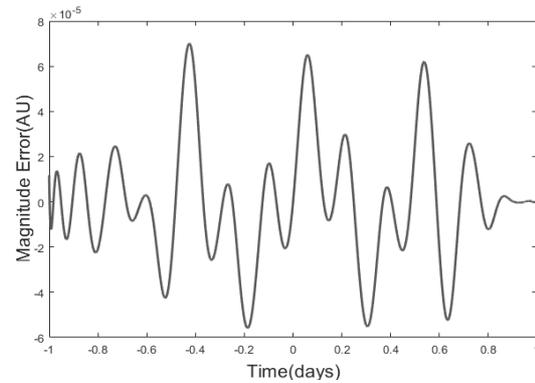


Fig. 7. Reconstruction error of magnitude

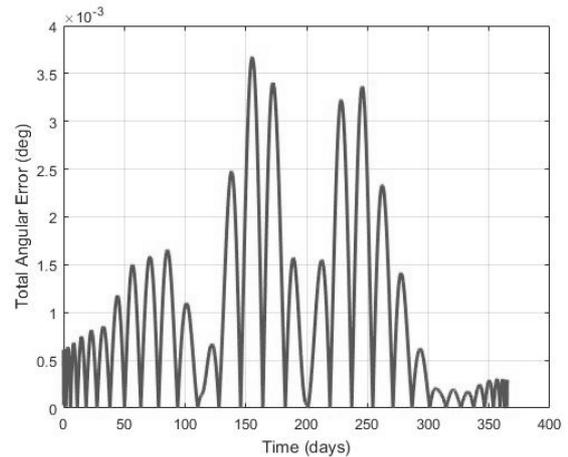


Fig. 8. Total angular error

Figure 8은 체비셰프 다항식으로 재생한 결과와 DE-430에서 얻은 태양 벡터와의 차이를 나타낸 것이다. 오차의 평균값은 0.0011896 degree(4.28256 arcsec)로 Vallado의 저서에서 제공하는 태양 벡터의 정밀도인 0.23 degree보다 정밀도가 뛰어난 것을 확인할 수 있다.

3.3 Performance

성능을 분석하기 위해 체비셰프 다항식 차수를 변경해가며 정밀도를 계산해보았고 기존의 천문력 생

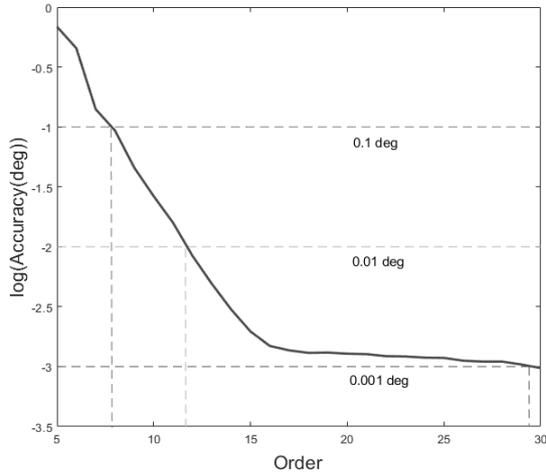


Fig. 9. Accuracy according to order

성식과의 계산시간을 비교해 보았다.

체비셰프 Curve fitting은 적은 데이터를 사용하여 함수를 구성하며, 식 (6)과 같이 압축하기 때문에 차수를 낮추는 것에 따라 정밀도가 달라진다. 2018년 1월 1일부터 2019년 1월 1일 기간에 대해 차수를 5에서 30까지 변경해 가며 정밀도를 비교해 본 결과는 Fig. 9와 같다. 점선으로 0.1, 0.01, 0.001 degree를 표시하였고, 차수 N(order)이 8 이상일 때 0.1 degree 이하의 정밀도를 가지며, 12 이상일 때 0.01 degree 이하의 정밀도를 가지는 것을 확인할 수 있다.

Figure 10에서는 Vallado의 천문력 생성식과 본 논문에서 제시하는 체비셰프 다항식의 계산 시간을 비교한 결과이다. 차수가 증가함에 따라 계산시간이 조금씩 더 길어지는 것을 확인할 수 있으며 주황색 그래프는 Vallado's method의 결과로 차수에 따른 결

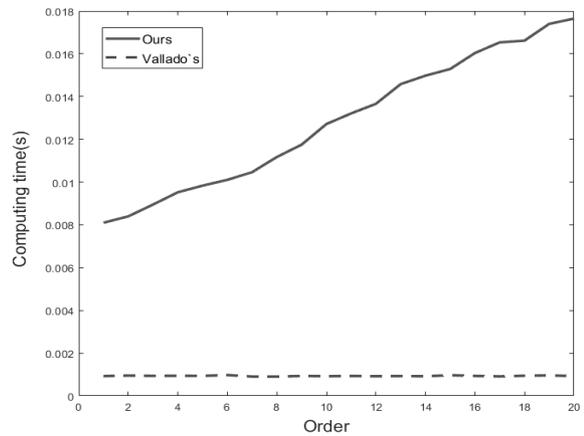


Fig. 10. Computation time

과는 아니지만 평균적으로 0.001sec의 결과가 나왔다. 계산 프로그램과 컴퓨터 성능에 따라 결과가 다를 수는 있지만 자세 결정을 위한 sensor sampling time을 10Hz라 가정했을 때 OBC에 사용하기에 충분할 것으로 판단된다.

IV. Planetary Ephemeris Generation Program

4.1 Development of Planetary Ephemeris Generation Program

위와 같이 체비셰프로 Curve fitting하여 태양 행성을 계산하는 방식을 MATLAB® Figure기반 앱인 GUIDE를 사용하여 천문력 생성 프로그램(Ephemeris Generation Code Generator, EGCG)을 개발하였다.

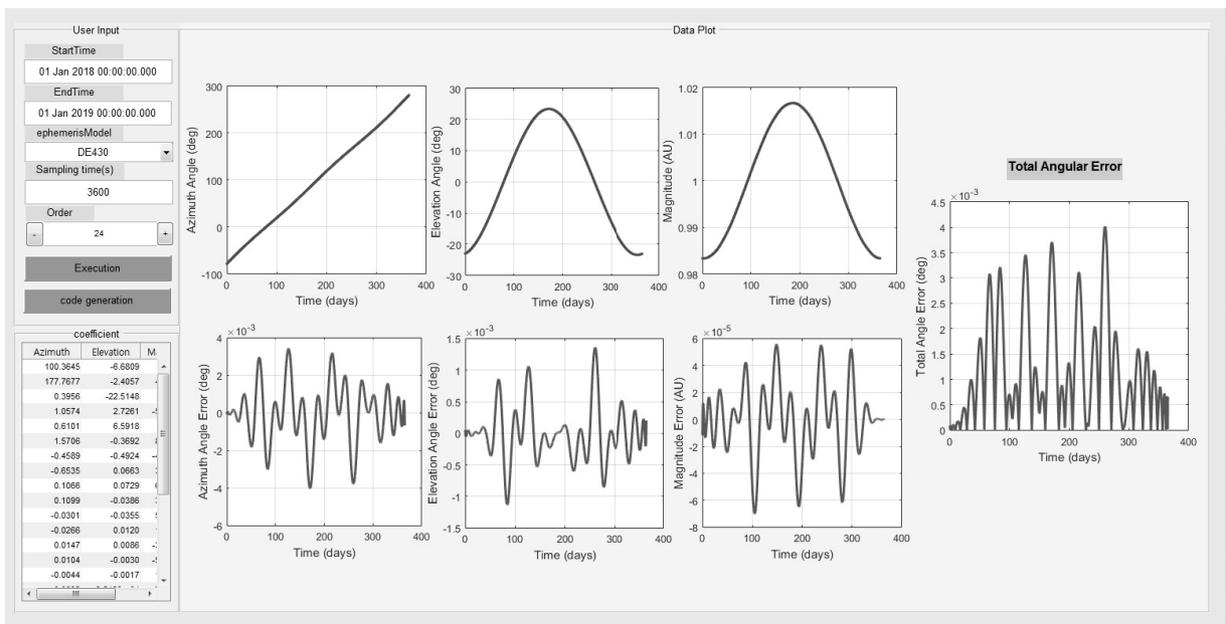


Fig. 11. Ephemeris Generation Code Generator

Figure 11에서 볼 수 있듯이 프로그램 입력으로는 시작시점, 임무 종료 시점, 'DE (number)', 시간 간격 그리고 체비셰프 방정식 차수가 있다. MATLAB® 함수 planetEphemeris는 입력으로 Julian date를 사용하기 때문에 사용자의 편의를 위해 프로그램 내부에서 시간 변환을 하도록 하여 사용자는 임무 시작 및 종료 시점을 UTC 시간 좌표계를 입력으로 사용할 수 있도록 설계하였다. ephemerisModel에서는 DE-Series 중 사용하고자 하는 reference 모델을 선택하였다. DE 시리즈의 각 모델마다 위치 정확도를 보장하는 기간이 다르기 때문에 중 405, 421, 423, 430, 432t 모델 중에 사용자가 이용하고자 하는 모델을 선택하여 사용할 수 있도록 하였다. Order 부분에서는 최종적으로 구해지는 체비셰프 방정식의 차수를 결정한다. 차수가 높을수록 더 높은 정밀도를 보장하지만 계산 시간이 더 늘어나는 단점이 있으며, 태양 센서의 경우 센서의 정밀도가 높지 않기 때문에 차수를 지나치게 증가시켜 정밀도를 높일 필요가 없다.

Execution을 눌러 프로그램을 실행하면 체비셰프 다항식의 계수를 계산하고 그 값들은 사용하여 재생한 결과를 위쪽 세장의 plot을 통해 확인할 수 있다. 그리고 아래의 사진을 통해 설정한 sampling time에 따른 planetEphemeris 함수의 출력과 재생한 결과를 비교한 오차값을 확인할 수 있다. 또한 coefficient 창에서는 프로그램을 통해 구한 체비셰프 다항식 계수를 볼 수 있다. 프로그램 실행이 완료되면 천문력 다항식 차수, 평균 오차, 최대오차를 Fig. 12와 같이 팝업으로 나타낸다.

최종적으로는 OBC에서 사용하기 위한 코드가 필요하므로 CPP파일을 생성하는 기능을 추가하였다. Fig. 11에서 볼 수 있는 code generation을 누르게 되면 Fig. 13과 같이 체비셰프 다항식과 각 계수가 포함되어있는 함수 파일을 생성하게 된다. 재생한 결과는 식 (10)~(12)를 통해 태양 벡터로 표현되고, 이를 기준 태양벡터로 사용할 수 있다.

$$x = \cos(el) * \cos(az) * mag \quad (10)$$

$$y = \cos(el) * \sin(az) * mag \quad (11)$$

$$z = \sin(el) * mag \quad (12)$$

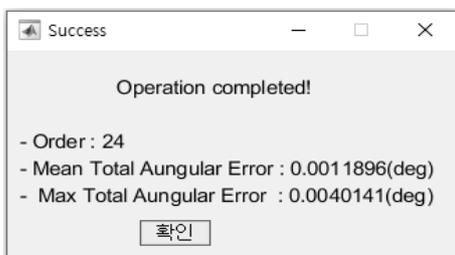


Fig. 12. Execution report

```
#include "SUNPOS.h"

double EpochMJD = 2.458119500000000e+06; // Epoch Time is 01 Jan 2018 0
double EndMJD = 2.458484500000000e+06; // End Time is 01 Jan 2019 00:0
double Sampling = 4.166666666666666e-02; // Sampling time

double b_az[25] = {1.003857778773564e+02,1.777899222652925e+02,3.97410273110
double b_el[25] = {-6.682808398171272e+00,-2.409341091859469e+00,-2.251298069
double b_mag[25] = {9.950426673694546e-01,4.469892978601172e-04,-1.6281122161

void Position_SUN(double SunVec[3], double ElapseTime)
{
    double Time = EpochMJD + ElapseTime;
    double Sun_az = CHEBYSHEV_Poly(b_az, EpochMJD, EndMJD-Sampling, Time)*DTR;
    double Sun_el = CHEBYSHEV_Poly(b_el, EpochMJD, EndMJD-Sampling, Time)*DTR;
    double Sun_mag = CHEBYSHEV_Poly(b_mag, EpochMJD, EndMJD-Sampling, Time);

    SunVec[0] = cos(Sun_el)*cos(Sun_az)*Sun_mag*AU;
    SunVec[1] = cos(Sun_el)*sin(Sun_az)*Sun_mag*AU;
    SunVec[2] = sin(Sun_el)*Sun_mag*AU;
}
```

Fig. 13. Result of Code Generation

V. 결론

DE-Series나 VSOP와 같은 정밀 천문력보다는 정밀하지 않지만 Vallado의 저서에서 제공하는 천문력 보다는 더 나은 정밀도를 가지면서 빠른 연산이 가능한 천문력을 생성하기 위해 체비셰프 다항식을 통한 Curve fitting을 수행하였다. 또한 사용자의 편의성을 위해 MATLAB®의 GUI를 활용한 천문력 생성 프로그램을 개발하였다. Vallado's method보다 높은 정밀도를 가질 뿐 아니라 막대한 데이터를 사용하여 빠른 연산이 불가능하거나 TDB를 사용하여 시간 좌표계 변환이 별도로 필요로 하는 등의 기존의 문제를 해결하였다. 본 논문에서 제안한 프로그램은 향후 고성능, 고정밀 임무를 수행하는 초소형위성에 탑재하여 사용하는데 유리할 것으로 사료된다.

후 기

본 논문은 'AI 기반 랑데부/도킹 기술검증용 위성 개발' 연구과제의 일부로써 수행되었으며, 이에 항공 우주연구원의 지원에 감사드립니다.

References

- 1) Folkner, W. M., Williams, J. G., Boggs, D. H., Park, R. S., and Kuchynka, P., "The planetary and lunar ephemerides DE430 and DE431," *Interplanetary Network Progress Report 196*, 2014.
- 2) Vallado, D. A., "Fundamentals of astrodynamics and applications," *Springer Science & Business Media*, Vol. 12., 2001.
- 3) Bretagnon, P., and Francou, G., "Planetary theories in rectangular and spherical variables-VSOP

87 solutions," *Astronomy and Astrophysics* 202, 1988, pp.309~315.

4) Meeus, J. H., "Astronomical algorithms. Willmann-Bell," Incorporated, 1991, ISBN 0-943396-35-2.

5) Kim, Y. D., Bang, H. C., and Kim, J. H., "Sensors and Actuators of Satellite Attitude Control," *Journal of the Institute of Control, Automation and Systems Engineer*, Vol. 3, No. 3, 1997, pp.29~35.

6) Lee, S. H., Yu, J., Lim, Y. C., and Gwak, H. K., "A Satellite Attitude Compensation Scheme Using Sun Sensor," *Journal of the Institute of Control, Automation and Systems Engineer*, Vol. 13, No. 7, 2007, pp.703~710.

7) Daffalla, M. M., TagElsir, A., and Kajo, A. S., "Hardware selection for attitude determination and control subsystem of 1U cube satellite," *Computing, Control, Networking, Electronics and Embedded Systems Engineering (ICCNEEE), International Conference on. IEEE*, 2015, pp.118~122.

8) Shuster, M. D., and Oh, S. D., "Three - Axis Attitude Determination from Vector Observations," *Journal of Guidance, Control and Dynamics*, Vol. 4, No. 1, 1981, pp.70~77.

9) O'Keefe, S. A., "Autonomous Sun-Direction Estimation Using Partially Underdetermined Coarse Sun Sensor Configurations," Ph.D., Department of Aerospace Engineering Sciences, University of Colorado, 2015, p.175.