

[항공우주시스템설계 Final Report]

2022년 2 월

달 탐사 궤도에서 행성과 별을 이용한 천체항법
Celestial Navigation Using Stars and Planets
on Lunar Exploration Orbit

2014-11238

김재우

지 도 교 수 : 기 창 돈

2022. . .

서울대학교 공과대학 기계항공공학부

달 탐사 궤도에서 행성과 별을 이용한 천체항법

Celestial Navigation Using Stars and Planets on Lunar Exploration Orbit

지도교수 : 기 창 돈

이 논문을 공학사 학위 졸업논문으로 제출합니다.

2022년 월 일

서울대학교 공과대학 기계항공공학부

2014-11238 김재우

위 논문을 공학사 졸업논문으로서 _____로 인정한다.

2022년 월 일

심사교수 _____ (인)

달 탐사 궤도에서 행성과 별을 이용한 천체항법

Celestial Navigation Using Stars and Planets on Lunar Exploration Orbit

기계항공공학부

2014-11238

김재우

요약

우주산업 시장의 팽창이 효율적인 우주탐사시스템의 가치를 견인하면서, 우주탐사 분야는 새로운 국면을 맞이하고 있다. 그와 동시에, 국제우주정거장(ISS, International Space Station)의 은퇴와 함께 달이 새로운 우주탐사의 거점으로 지목되면서, 효율적인 달 탐사 시스템을 구축하는 것은 인류 역사상 그 어느때보다 큰 경제적 가치를 내포하게 되었다. 이러한 사회·경제적 배경에 발맞추어 이 연구에서는 달 탐사 궤도에 활용될 수 있는 효율적인 항법시스템을 제시한다. 이 항법시스템은 모든 우주비행체에 탑재되는 자세결정센서인 별 추적기(star tracker)와 행성 궤도력(planets ephemeris)을 활용하여 추가적인 장치를 필요로 하지 않는다. 이에 대한 내용은 이미 선행연구된 바가 있지만, 이 연구에서는 이를 단순화하여 항법알고리즘이 가지는 수학적인 의미를 명확히 하고, 이를 통해 항법의 성능에 지배적인 영향을 미치는 요소들을 규명한다. 특히, '치명적 오차'라는 개념을 도입하여 항법의 안정성에 치명적인 영향을 줄 수 있는 상황을 규명한다. 또한, 이를 검증하기 위해 시뮬레이션을 진행한다. 시뮬레이션은 행성들의 임의배치를 통해 항법알고리즘의 성능을 검증하는 시뮬레이션1, 이를 바탕으로 실제 달 탐사 궤도 위에서 성능을 검증하는 시뮬레이션2으로 구성된다.

I. 서론

새롭게 등장한 우주분야 사기업들이 비즈니스 모델을 발굴하고 이를 통해 이익을 창출해내면서, 우주탐사는 과거 국가적 위상을 드높이는 역할을 넘어 새로운 시장을 만들고 사회 전체의 가치를 빠르게 키워나가고 있다. 또한, 달에 인간이 지속적으로 머무를 수 있는 환경을 조성하고, 유인화성탐사를 실현하고자 하는 목적을 가지고 진행중인 NASA의 유인 달 탐사 프로젝트(The Artemis Program)는 우주환경을 인간의 생활권 안으로 편입시키려는 시도를 하고 있다. 이러한 상황들을 고려해 보았을 때, 앞으로 우주탐사는 과거보다 더욱 빈번하게 이루어질 것을 기대해볼 수 있으며, 특히 달은 심우주탐사의 거점지로 활용될 가능성이 높아보인다. 이에 따라 비용효율적인 달 탐사 시스템을 연구하고 구축하는 일은 앞으로 더 높은 효용을 가질 것으로 기대된다.

항법시스템(navigation system)이란 사용자의 위치정보를 제공하는 시스템이다. 사용자가 위치정보를 파악할 수 있어야 계획했던 경로대로 진행되고 있는지, 아니라면 어떻게 제어해야 목적지에 성공적으로 도달할 수 있는지 파악하고 의사결정을 할 수 있기 때문에 우주임무를 수행하기 위해서는 필수불가결한 요소이다. 따라서 모든 우주 발사체 및 탐사선은 항법시스템을 포함하고 있으며, 항법시스템으로 가장 흔하게 채택되는 것은 관성항법시스템(INS, Inertial Navigation System)이다. 관성항법시스템은 내부의 자이로를 통해 얻은 가속도 데이터를 적분하여 사용자가 속도와 위치정보를 얻을 수 있도록 한다. 따라서 관성항법시스템은 지상국으로부터 전파 송수신이 어려운 심우주 공간에서도 외부 환경에 무관하게 위치정보를 얻을 수 있다는 강점이 있다. 하지만, 가속도를 시간에 대해 적분하여 위치정보를 얻기 때문에, 시간에 대해 오차가 누적된다는 문제가 있고, 따라서 이를 보정해줄 수 있는 보조항법시스템이 필수적이다.

모든 우주발사체 및 탐사선은 자세결정을 위한 시스템을 갖추고 있는데, 가장 흔하게 사용되는 장비는 별 추적기(star tracker)이다[1][2][3]. 별 추적기는 값싸고, 가볍지만 그 정확도가 arcsec 단위로 매우 정밀한 것으로 알려져 있어 효율적인 자세결정시스템이다. 별 추적기는 카메라와 마이크로컴퓨터로 이루어진 장비이다. 카메라는 일종의 센서로, 외부의 시각정보를 시간에 대해 독립적으로 획득한다. 마이크로컴퓨터는 메모리에 미리 저장된 성표(star catalogue)와 식별알고리즘(star identification algorithm), 여타 유용한 데이터와 프로그램, 그리고 카메라로부터 얻어낸 시각정보를 처리하여 유용한 정보를 사용자에게 제공한다. 이 때, 카메라는 외부의 시각정보를 시간에 대해 독립적으로 얻어낸다. 이러한 별 추적기의 경제성과 성능, 시간에 대한 독립성을 생각해보았을 때, 별 추적기를 이용하여 항법시스템을 구축할 수 있다면 관성항법시스템의 보조 역할을 효율적으로 해낼 것이며, 독립적인 항법시스템으로도 그 의미가 있다.

별 추적기를 통해 얻은 시각정보로 항법을 하는 것은 사람이 두 눈을 이용해 자신의 위치정보를 추정하는 것과 유사하다. 사람은 자신의 두 눈이 하나의 대상과 이루는 각, 또는 여러개의 대상들과 자신이 이루는 각을 인지하고, 물체가 얼마나 떨어져있는지에 대한 정보, 즉 물체와의 상대적인 위치정보를 추정할 수 있다. 비슷한 방법으로 별 추적기를 이용하여 여러개의 천체를 관측하고 관측자를 중심으로하는 천체들의 방향정보를 얻어낼 수 있다면 관측자의 상대적인 위치정보를 추정할 수 있을 것이다. 또한, 이 때 관측된 천체들의 기준좌표계상에서의 위치를 알고 있다면, 관측자의 기준좌표계 위에서의 위치정보를 구할 수 있을 것이다.

앞서 언급한 내용은 선행연구가 진행된 바 있다[4][5]. 앞선 연구에서는 행성-행성 궤도에서 별 추적기와 최소

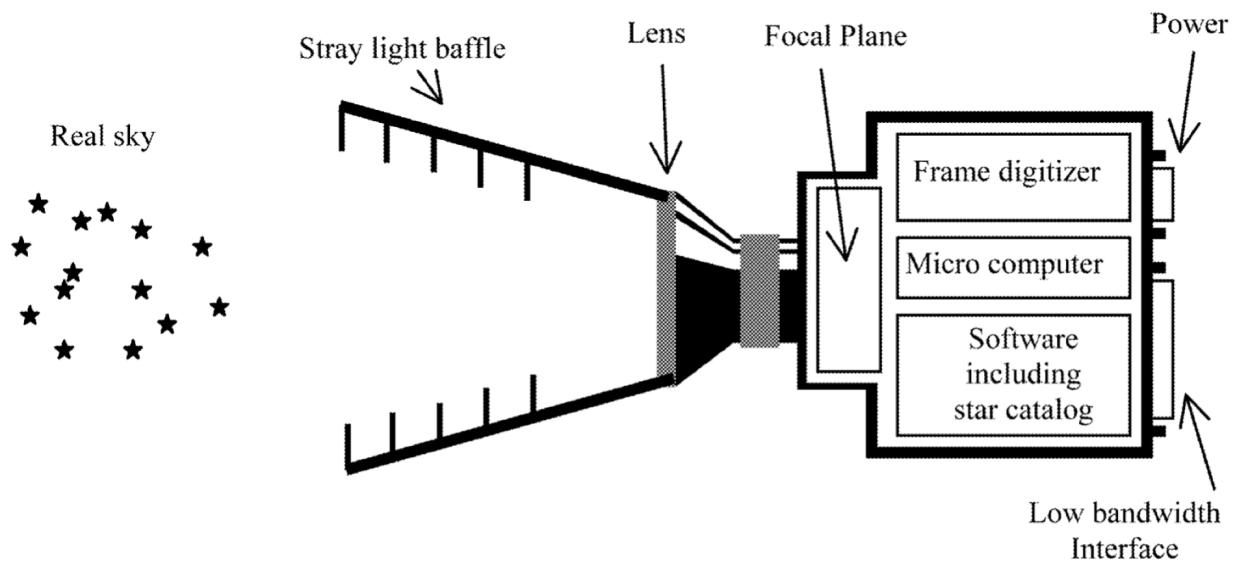


Fig. 1 별 추적기 [3]

제곱법을 활용한 항법알고리즘을 제시하고, 그 알고리즘의 성능을 해석적, 수치적으로 제시하였다. 이 연구에서는 위의 알고리즘을 달 탐사 궤도 위, 2차원 평면에서 단순화하여 해석한다. 단순화된 문제 해결을 통해 항법 성능에 결정적인 영향을 미치는 요소들을 구체적으로 밝힌다. 특히, ‘치명적 오차’라는 개념을 도입하여 치명적 오차에 영향을 미치는 요인과 치명적 오차가 항법의 성능에 미치는 영향을 시뮬레이션을 통해 발견하고, 해석하며 그 해석 내용을 검증한다.

II. 문제정의

A. 행성 궤도력(planetary ephemeris) 및 국제표준천체좌표계(ICRF, International Celestial Reference Frame)[6][7]

문제정의에 앞서, 이 연구에서 사용하는 행성 궤도력 데이터와 기준좌표계를 설명한다. 행성 궤도력은 시간에 대한 행성들의 위치정보를 미리 계산해놓은 것이다. 이 연구에서는 NASA JPL(Jet Propulsion Laboratory)에서 제공하는 DE405를 이용할 것이다. DE405는 1997년 5월에 제작되어 널리 사용되고 있는 궤도력 중하나이며, 1599년 12월 9일 2201년 2월 20일까지의 천체의 위치정보를 얻을 수 있다. 정확도는 내행성의 경우 $0.001''$, 외행성의 경우 $0.1''$ 이하이다. 이는 차후 시뮬레이션에서 도입할 별 추적기의 오차 $1^\circ, 10''$ 를 고려하면 무시할만한 수준이다. 이 연구에서는 행성 궤도력 데이터의 오차를 고려하지 않는다.

DE405는 ICRF1(International Celestial Reference Frame 1)을 기준좌표계로하여 작성되었다. ICRF는 국제적으로 천체를 다룰 때 사용되는 좌표계의 표준이다. 원점은 태양계의 질량중심이며, 매우 먼 거리에 있는 별들이 발산하는 전자기파를 이용하여 축을 아주 정밀하게 설정한다. 이 때, 각 축의 방향은 지구적도좌표계와 거의 동일하다. 각

축의 정밀도는 $0.001''$ 미만의 단위로, 궤도력과 마찬가지로 시뮬레이션에서 가정할 별 추적기의 각 오차에 비해 매우 작다. 이 연구에서는 좌표계의 축오차를 고려하지 않는다.

B. 문제정의

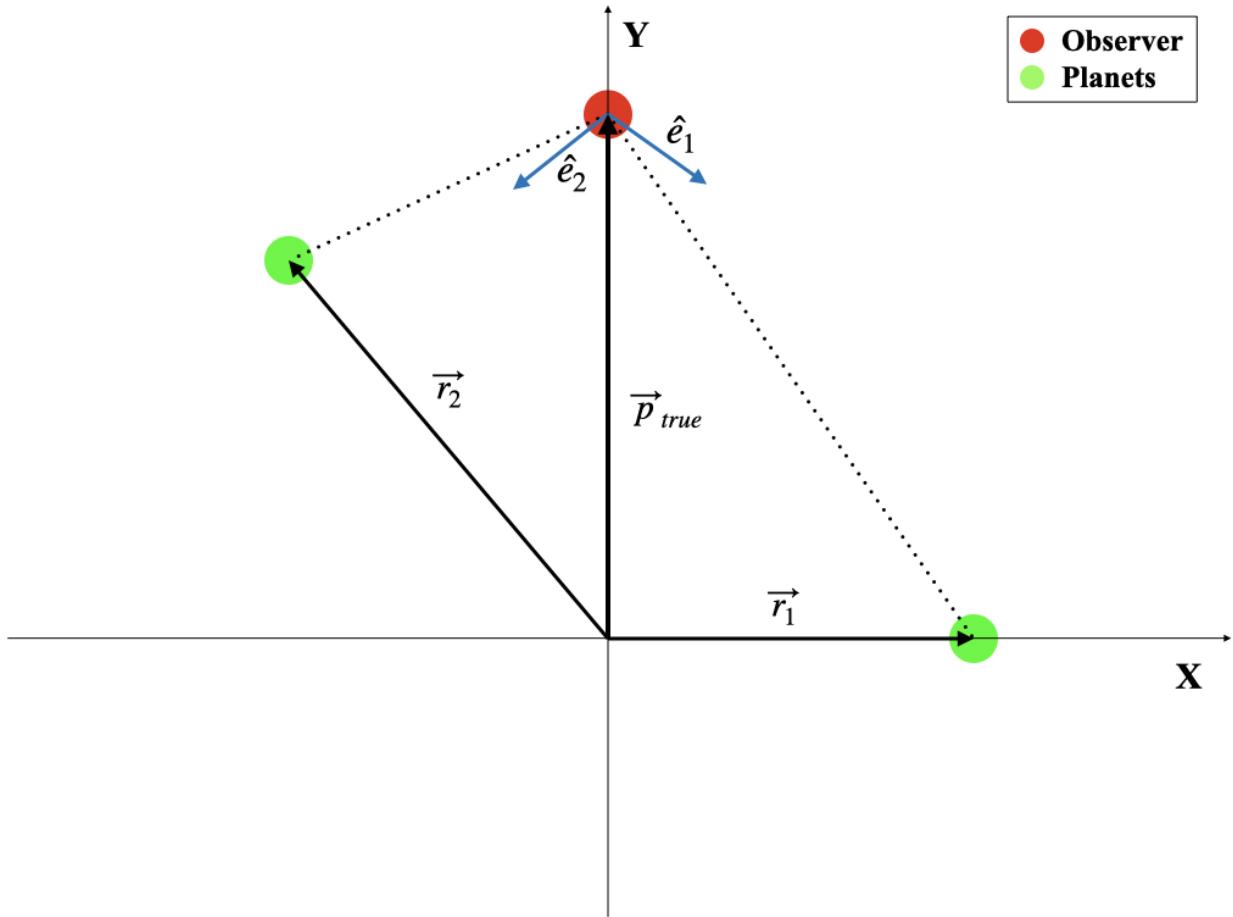


Fig. 2 문제상황

이 연구에서는 단순화를 위해 2D 평면 위에서 문제를 정의할 것이다. 별 추적기를 통해 n 개의 행성이 관측된 상황을 가정하며, 이 때 각 행성들은 정확히 식별 가능하다고 가정한다. 또한, 자세정보는 별 추적기를 통해 정확히 얻을 수 있다고 가정하여 고려대상에서 제외한다. Fig. 2은 이 연구에서 다룰 문제상황을 간단하게 나타낸 것이다. 정리하면,

별 추적기를 통해 측정된 n 개의 관측자-행성 방향벡터 : $\{\hat{e}_1, \hat{e}_2, \dots, \hat{e}_n\}$

행성 궤도력을 통해 얻은 n 개의 행성의 기준좌표계 위에서의 위치벡터 : $\{\vec{r}_1, \vec{r}_2, \dots, \vec{r}_n\}$

을 이용하여 관측자의 위치벡터 \vec{p}_{true} 를 추정하는 것이 이 연구에서 다룰 문제이다.

III. 위치추정 알고리즘[4][5]

A. 위치추정 알고리즘1 - 최소제곱법(Least Squares)

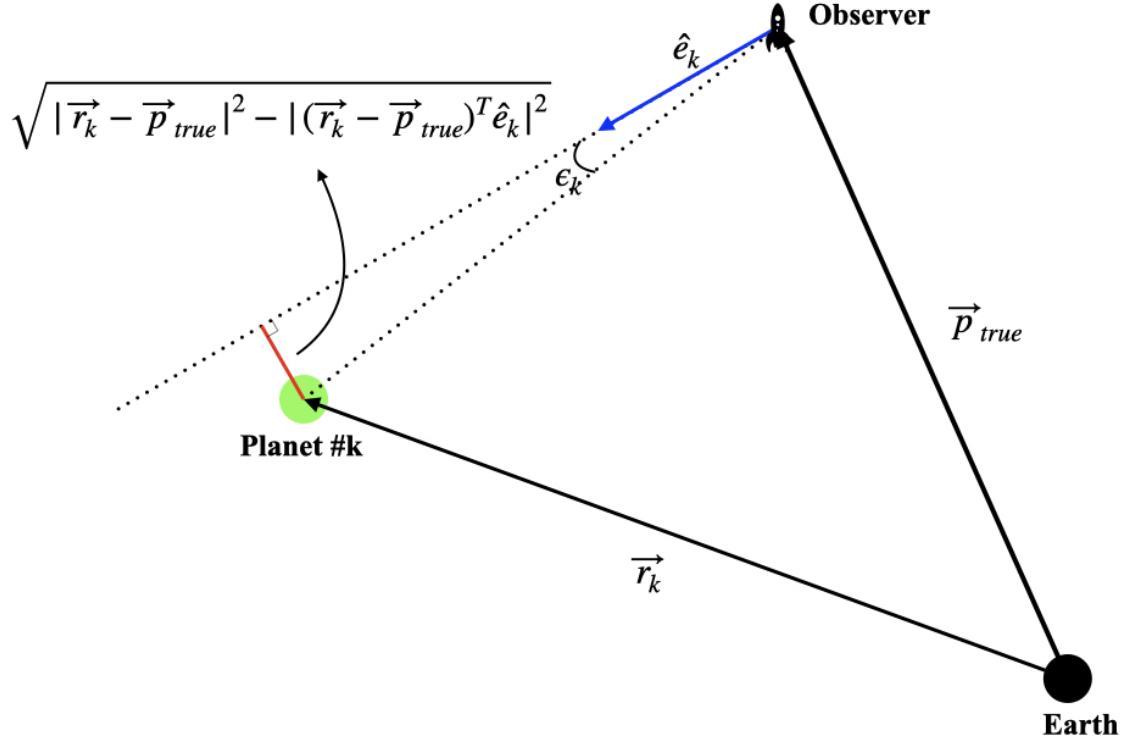


Fig. 3 k번째로 관측된 행성과 측정치 오차

별 추적기를 통해 관측된 n개의 행성 중 k번째로 관측된 행성을 생각한다. 측정된 관측자-행성 방향벡터 \hat{e}_k 는 각오차 ϵ_k 을 포함한 값이다. 이 때, \hat{e}_k 방향에 내린 관측자-행성을 잇는 벡터의 수선의 발의 크기(Fig. 3의 빨간색 선분의 길이)는 각오차가 커짐에 따라 함께 커질 것이다. 이에 착안하여 관측된 각 행성들의 관측치에서 발생한 오차를 $\sqrt{|\vec{r}_k - \vec{p}_{true}|^2 - |(\vec{r}_k - \vec{p}_{true})^T \hat{e}_k|^2}$ 로 정하고, 관측된 n개의 행성들의 오차의 제곱합을 목적함수로 정한다.

$$L = \sum_{k=1}^n (|\vec{r}_k - \vec{p}|^2 - |(\vec{r}_k - \vec{p})^T \hat{e}_k|^2) \quad (1)$$

이 때, 목적함수는 \vec{p} 에 대한 함수이다. 이 함수값을 최소로 만드는 최적해 \hat{p} 를 위치추정벡터로 정한다. 목적함수는 p 에 대한 2차함수이고 항상 양수이므로 ($\because |(\vec{r}_k - \vec{p})^T \hat{e}_k| \leq |\vec{r}_k - \vec{p}|$), 최적해가 항상 존재하며, 그 값을 벡터미분법을

이용해 간단하게 구할 수 있다.

$$\frac{dL}{d\hat{p}}(\hat{p}) = -2 \sum_{k=1}^n [(\vec{r}_k - \hat{p}) - \hat{e}_k \hat{e}_k^T (\vec{r}_k - \hat{p})] = 0 \quad (2)$$

정리하면,

$$\hat{p} = \left(\sum_{k=1}^n \hat{E}_k \right)^{-1} \left(\sum_{k=1}^n \hat{E}_k \vec{r}_k \right), \text{ where } \hat{E}_k = I - \hat{e}_k \hat{e}_k^T \quad (3)$$

B. 최소제곱법의 기하학적 이해

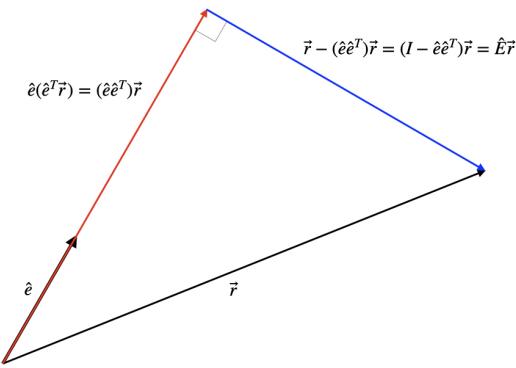


Fig. 4 변환 $\hat{e}\hat{e}^T$ 와 \hat{E}

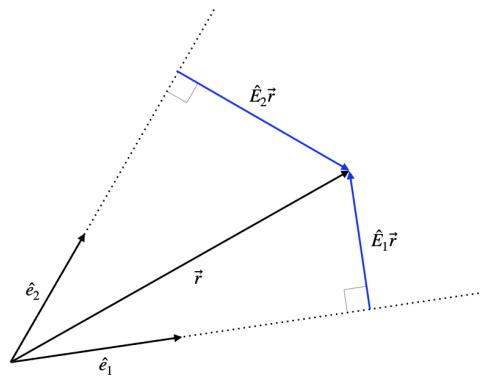


Fig. 5 $\hat{e}_1, \hat{e}_2, \vec{r}, \hat{E}_1 \vec{r}, \hat{E}_2 \vec{r}$ 의 관계

지금까지 최소제곱법을 이용한 항법알고리즘에 대한 아이디어를 설명하고 위치추정벡터를 구하는 과정을 간단히 소개하였다. 하지만, 이후 시뮬레이션의 방향성을 설정하고 그것의 결과를 분석하기 위해서는 항법알고리즘에 대한 좀 더 깊이있는 이해가 필요하다.

모든 행렬은 선형변환이다. $\hat{e}_k \hat{e}_k^T$ 는 2×2 행렬로, 임의의 벡터 \vec{r} 을 \hat{e}_k 위로 정사영하는 선형변환이다. 따라서, $\hat{E}_k = I - \hat{e}_k \hat{e}_k^T$ 는 임의의 벡터 \vec{r} 을 \hat{e}_k 위에 내린 수선의 발로 대응시키는 선형변환이다. Fig. 4는 이것을 그림으로 나타낸 것이다. Fig. 5는 한 예시로 $\hat{e}_1, \hat{e}_2, \vec{r}$ 와 $\hat{E}_1 \vec{r}, \hat{E}_2 \vec{r}$ 사이의 관계를 나타낸 것이다. $\hat{p} = (\sum_{k=1}^n \hat{E}_k)^{-1} (\sum_{k=1}^n \hat{E}_k \vec{r}_k)$ 를 구하는 과정은 각 행성들의 위치벡터를 별 추적기를 통해 측정된 관측자-행성 방향벡터 위로의 수선의 발들의 합에 $\sum_{k=1}^n \hat{E}_k$ 의 역변환을 취하는 것이다. 또한, $\sum_{k=1}^n \hat{E}_k \vec{p} = \sum_{k=1}^n \hat{E}_k \vec{r}_k$ 를 만족시키는 \hat{p} 를 찾는 과정으로 이해할 수도 있다.

행성이 2개 관측되었을 때를 살펴보자. 위의 방법을 사용하면, 각 행성을 지나며 측정된 방향벡터에 평행한 두 직선의 교점으로 위치추정벡터가 결정된다. 이 때 목적함수값은 0이 된다. 행성이 3개 이상 관측된 경우는 두 직선의 교점으로 최적해가 결정되지 않는다. Fig. 7과 Fig. 8의 검은 점선으로 표현된 \hat{p} 가 최적해일 때, 빨간 실선 벡터들의 합은 같다. 즉, 2개의 행성을 이용할 때와 비교해서 \hat{e}_k 에 수직한 방향들 사이에서 조정이 일어난다. 이 조정과정이 위치추정벡터에 어떤 영향을 미치는지를 관찰하는 것이 이후 시뮬레이션의 주요 목표 중 하나가 될

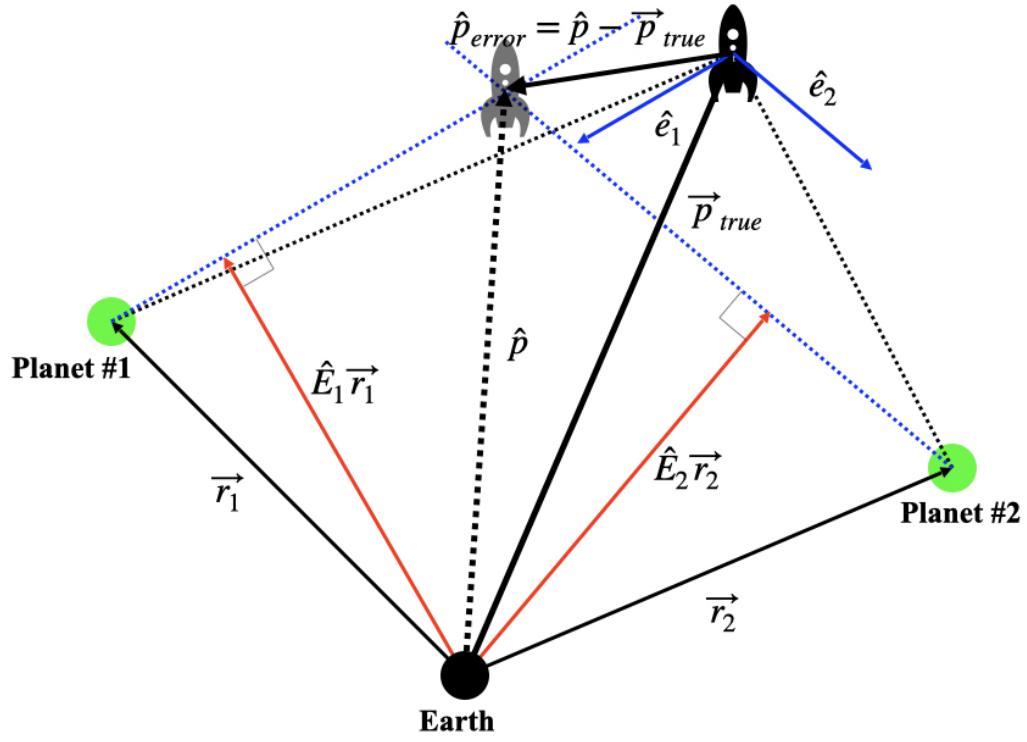


Fig. 6 2개의 행성이 관측된 상황

것이다.

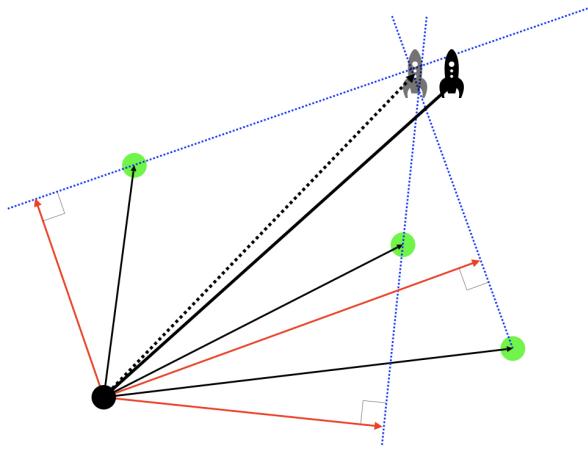


Fig. 7 $\sum \vec{E}_k \vec{r}_k$

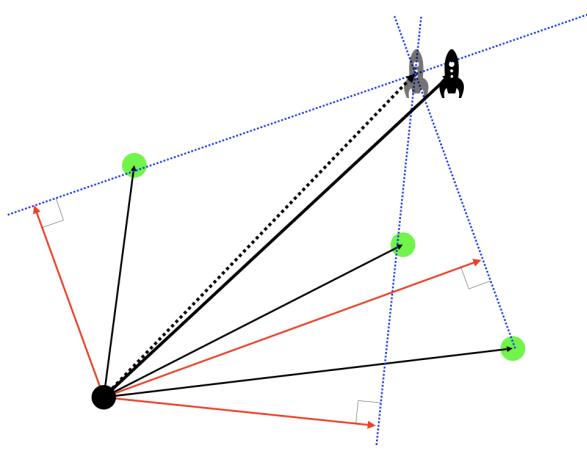


Fig. 8 $\sum \vec{E}_k \hat{p}$

C. 위치추정 알고리즘2 - 가중 최소제곱법(WLS, Weighted Least Squares)

각오차(ϵ)가 커짐에 따라, 측정된 관측자-행성 방향벡터 위로우 푸선의 발($\sqrt{|\vec{r} - \vec{p}|^2 - |(\vec{r} - \vec{p})^T \hat{e}|^2}$)이 함께 커지는 것에 착안하여 오차와 목적함수를 정의했다. 하지만 이 방식은 측정치에서 같은 각오차가 발생하더라도 더 먼거리에 있는 행성의 각오차가 목적함수에 더 많은 영향을 끼치게 된다(Fig. 9). 가령, A행성보다 100배 멀리 있는 B행성은 목적함수에 10000배의 영향력을 갖는다. 달 탐사 궤도에서 안정적으로 관측 가능한 행성사이의 거리 수준을 고려했을 때, 이 차이를 무시할 수 없다(Fig. 10). 따라서 거리차이에 의한 효과를 보정하기 위한 가중치를

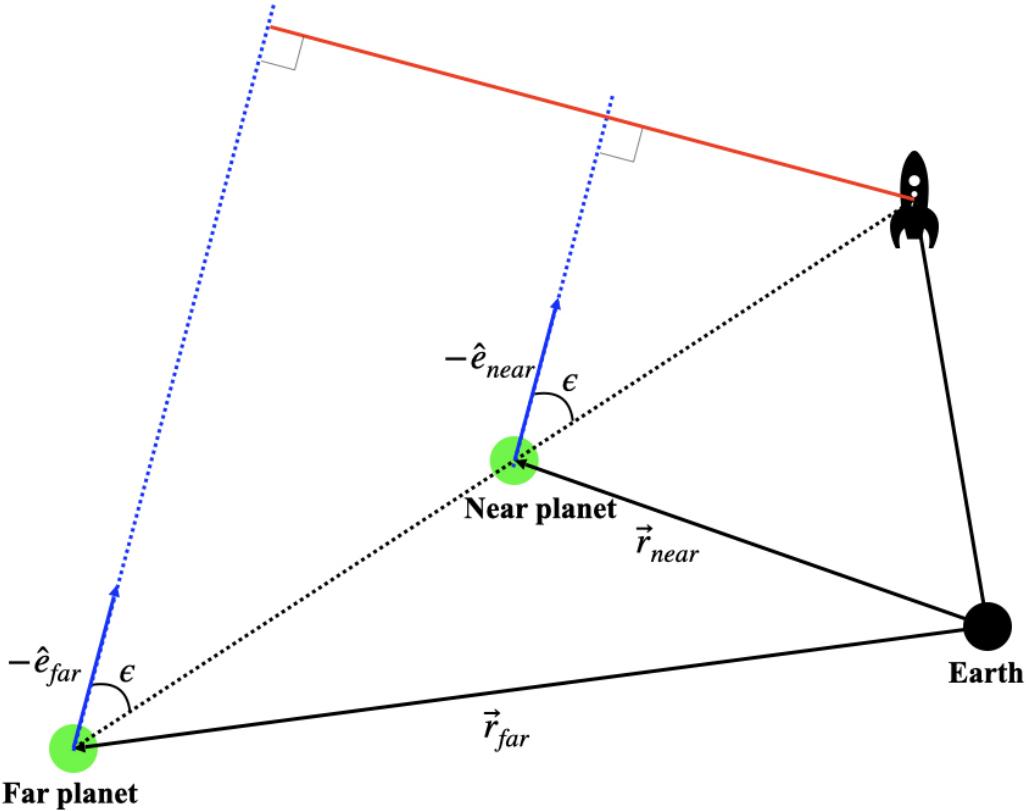


Fig. 9 면 거리와 가까운 거리에 있는 행성

설정해주어야 더 좋은 성능의 항법알고리즘을 구현할 수 있을 것이다. 측정치를 초 단위로 생성한다고 할 때, 관측자-행성 사이의 거리의 변화량은 그 값에 비해 매우 작은 값이다. 따라서, 이전시간에 구한 최적해 \hat{p}_{prev} 를 활용하여 $w_k = |\vec{r}_k - \hat{p}_{prev}|^{-2}$ 를 각 항의 가중치로 설정하면 추가적인 계산 없이 목적을 달성할 수 있을 것이라 판단하였다. 이에 따라 목적함수를 다음과 같이 재정의할 수 있다.

$$L = \sum_{k=1}^n [(|\vec{r}_k - \vec{p}|^2 - |(\vec{r}_k - \vec{p})^T \hat{e}_k|^2) w_k] \quad (4)$$

○ 때, 최적해는

$$\hat{p} = \left(\sum_{k=1}^n \hat{E}_k w_k \right)^{-1} \left(\sum_{k=1}^n \hat{E}_k \vec{r}_k w_k \right) \quad (5)$$

	MERCURY	VENUS	EARTH	MOON	MARS	JUPITER	SATURN	URANUS	NEPTUNE	PLUTO
<u>Mass</u> (10^{24} kg)	0.330	4.87	5.97	0.073	0.642	1898	568	86.8	102	0.0146
<u>Diameter</u> (km)	4879	12,104	12,756	3475	6792	142,984	120,536	51,118	49,528	2370
<u>Density</u> (kg/m ³)	5427	5243	5514	3340	3933	1326	687	1271	1638	2095
<u>Gravity</u> (m/s ²)	3.7	8.9	9.8	1.6	3.7	23.1	9.0	8.7	11.0	0.7
<u>Escape Velocity</u> (km/s)	4.3	10.4	11.2	2.4	5.0	59.5	35.5	21.3	23.5	1.3
<u>Rotation Period</u> (hours)	1407.6	-5832.5	23.9	655.7	24.6	9.9	10.7	-17.2	16.1	-153.3
<u>Length of Day</u> (hours)	4222.6	2802.0	24.0	708.7	24.7	9.9	10.7	17.2	16.1	153.3
<u>Distance from Sun</u> (10^6 km)	57.9	108.2	149.6	0.384*	227.9	778.6	1433.5	2872.5	4495.1	5906.4
<u>Perihelion</u> (10^6 km)	46.0	107.5	147.1	0.363*	206.6	740.5	1352.6	2741.3	4444.5	4436.8
<u>Aphelion</u> (10^6 km)	69.8	108.9	152.1	0.406*	249.2	816.6	1514.5	3003.6	4545.7	7375.9
<u>Orbital Period</u> (days)	88.0	224.7	365.2	27.3*	687.0	4331	10,747	30,589	59,800	90,560
<u>Orbital Velocity</u> (km/s)	47.4	35.0	29.8	1.0*	24.1	13.1	9.7	6.8	5.4	4.7
<u>Orbital Inclination</u> (degrees)	7.0	3.4	0.0	5.1	1.9	1.3	2.5	0.8	1.8	17.2
<u>Orbital Eccentricity</u>	0.205	0.007	0.017	0.055	0.094	0.049	0.057	0.046	0.011	0.244
<u>Oblliquity to Orbit</u> (degrees)	0.034	177.4	23.4	6.7	25.2	3.1	26.7	97.8	28.3	122.5
<u>Mean Temperature</u> (C)	167	464	15	-20	-65	-110	-140	-195	-200	-225
<u>Surface Pressure</u> (bars)	0	92	1	0	0.01	Unknown*	Unknown*	Unknown*	Unknown*	0.00001
<u>Number of Moons</u>	0	0	1	0	2	79	82	27	14	5
<u>Ring System?</u>	No	No	No	No	No	Yes	Yes	Yes	Yes	No
<u>Global Magnetic Field?</u>	Yes	No	Yes	No	No	Yes	Yes	Yes	Yes	Unknown
	MERCURY	VENUS	EARTH	MOON	MARS	JUPITER	SATURN	URANUS	NEPTUNE	PLUTO

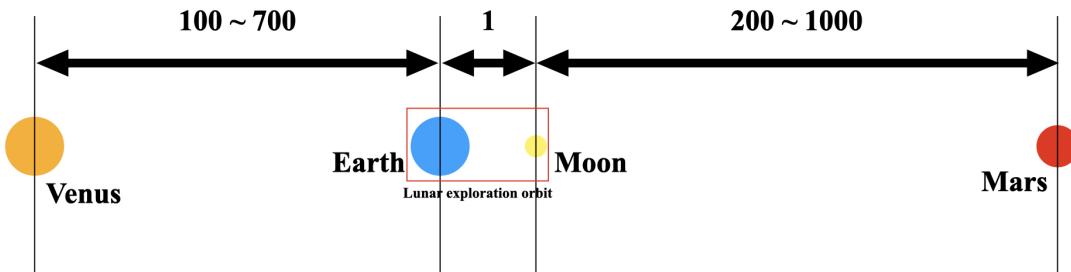


Fig. 10 달 탐사 궤도에서 안정적으로 관측 가능한 행성사이의 거리 수준[8]

IV. 시뮬레이션

앞서 정의한 항법알고리즘의 성능에 미치는 요인을 관찰하기 위한 시뮬레이션을 설계하고 진행한다. 먼저, 시뮬레이션1에서는 관측과 행성들 사이의 거리와 각도가 항법알고리즘의 성능에 어떤 영향을 미치는지 관찰하기 위해 임의로 대상들을 배열한 시뮬레이션을 진행한다. 이후, 그 결과를 통해 시뮬레이션2에서 실제 달 탐사궤도(호만 전이궤도)에서 어떤 성능을 보일 것인지 예측하고, 예측과 실제 결과가 동일한지 확인하는 시뮬레이션을 진행한다.

A. 시뮬레이션1 - 임의의 배열

임의의 배열로 시뮬레이션을 진행할 때, 공통사항은 다음과 같다.

- 달 탐사궤도에서 안정적으로 관측할 수 있는 행성(지구, 달, 금성, 화성)의 거리수준을 고려하여 시뮬레이션을 진행한다. 이 때, 안정적으로 관측 가능한 행성의 설정은 지구와의 거리, 이심율, 주기 등을 고려하여 선정하였다.
- 단순화를 위해 2D 평면 위에서 진행한다.
- 거리의 단위는 무차원으로 진행한다.
- 별 추적기를 통해 측정한 관측자-행성 방향벡터의 각오차는 표준편차가 1° , 평균이 0인 정규분포를 따르는 것으로 가정하여 진행한다.
- 각 상황에 대해 5000번의 몬테카를로 시뮬레이션을 진행한다.

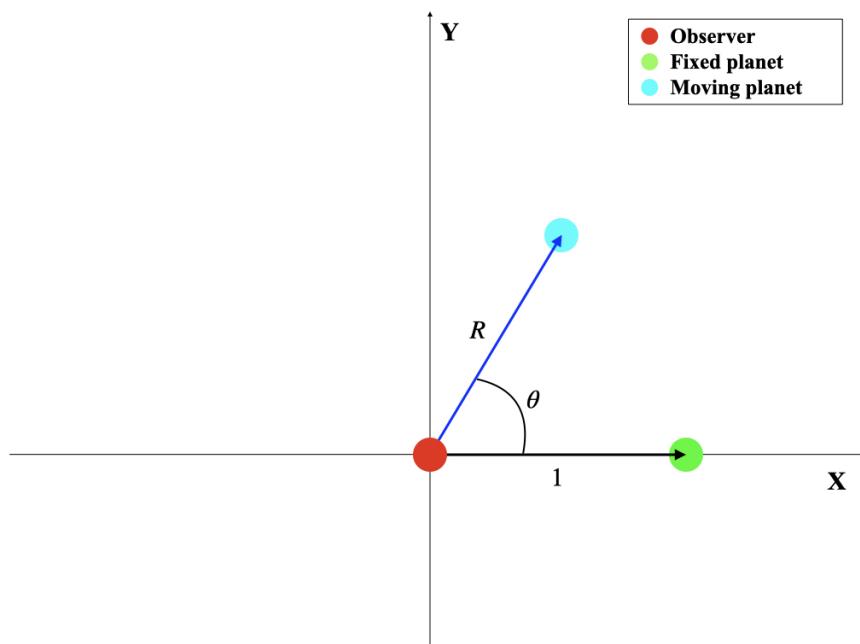


Fig. 11 Case #1

1. <Case #1 - 2개의 행성 관측>

2D 평면 위에 행성 하나를 $(1, 0)$ 에 고정한 후, 나머지 행성 하나를 거리(R)와 방향(θ)을 바꾸어가며 위치추정 벡터를 관찰한다(Fig. 11). 2개의 행성을 이용하여 위치추정벡터를 생성할 경우 LS와 WLS의 해가 동일하므로, 더 간단한 LS를 위치추정 알고리즘으로 채택한다.

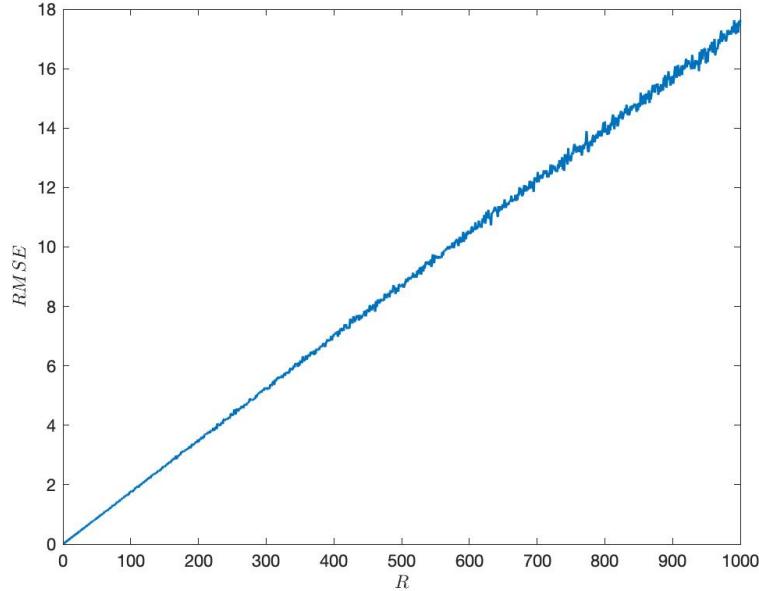


Fig. 12 Case #1의 결과 : R vs RMSE, $\theta = 90^\circ$

R 이 커짐에 따라 선형적으로 오차의 크기가 증가하는 것을 확인할 수 있었으며, $\theta = 90^\circ$ 일 때 가장 좋은 성능을 보였다. 위치추정치의 시각화를 통해 지상에서 2개의 VOR station을 이용하여 위치추정치를 생성할 때와 같은 모양을 보이는 것을 확인할 수 있다(Fig. 14). 추가적으로, $\theta = 0^\circ, 180^\circ$ 부근, 즉, 관측자와 행성들이 일직선상에 놓일 때 오차가 기하급수적으로 커지는 것을 관찰할 수 있었고, 이것에 대한 설명이 필요하다고 판단하였다.

$\theta = 180^\circ$ 인 상황에서(Fig. 15), $3 - \sigma - line$ 을 벗어나는 지점에 위치추정치가 다수 발생한 것을 확인할 수 있다. 이런 위치추정치는 Fig. 16에서처럼 관측자-행성 방향벡터 측정치의 각오차가 서로 엇갈려서 발생했을 때 생성된다. 이 때, 오차의 크기는 그렇지 않은 경우보다 매우 크게 발생한다. 따라서 안정적인 항법을 위해서는 이 특수한 상황에 대한 관찰이 필요해보인다. 이 연구에서는 앞으로 이것을 특별히 '치명적 오차'라 정의한다.

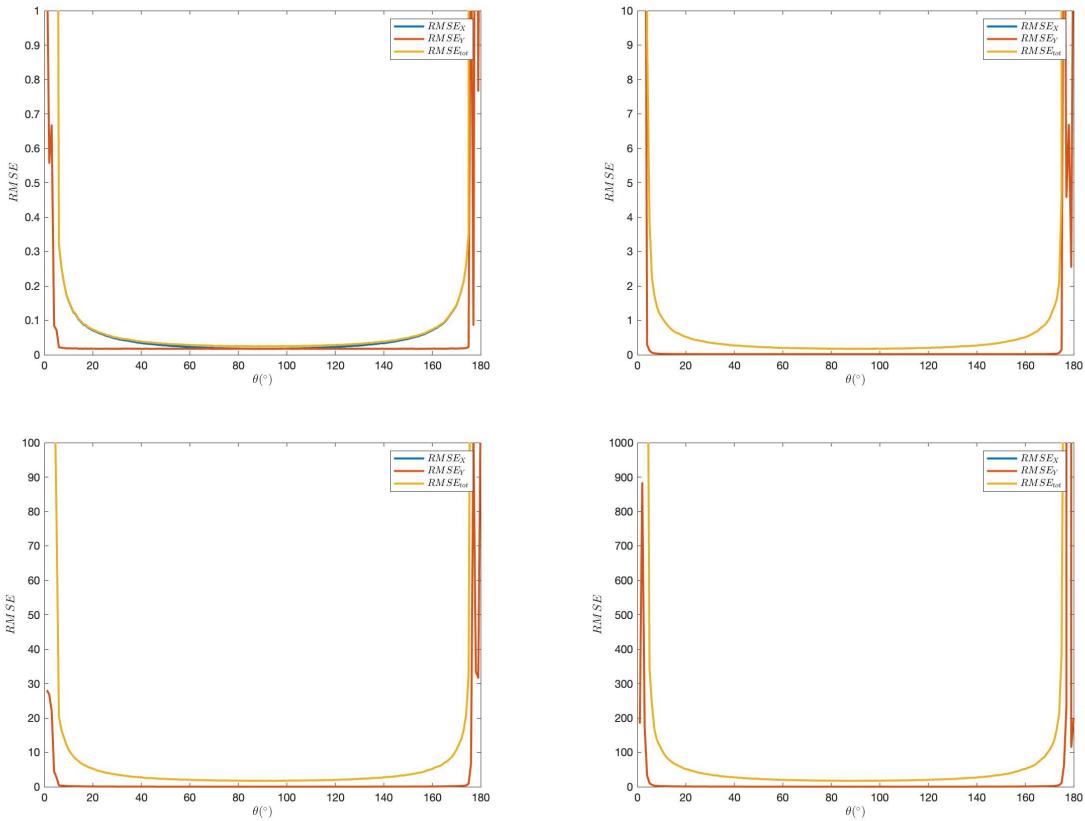


Fig. 13 Case #1의 결과 : θ vs RMSE 그래프, 왼쪽 위부터 차례대로 $R = 1, 10, 100, 1000$

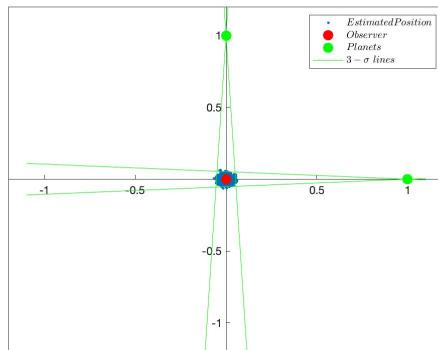


Fig. 14 Case #1의 결과 : $R = 1, \theta = 90^\circ$

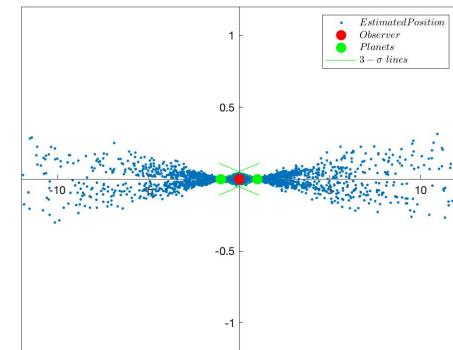


Fig. 15 Case #1의 결과 : $R = 1, \theta = 1800^\circ$

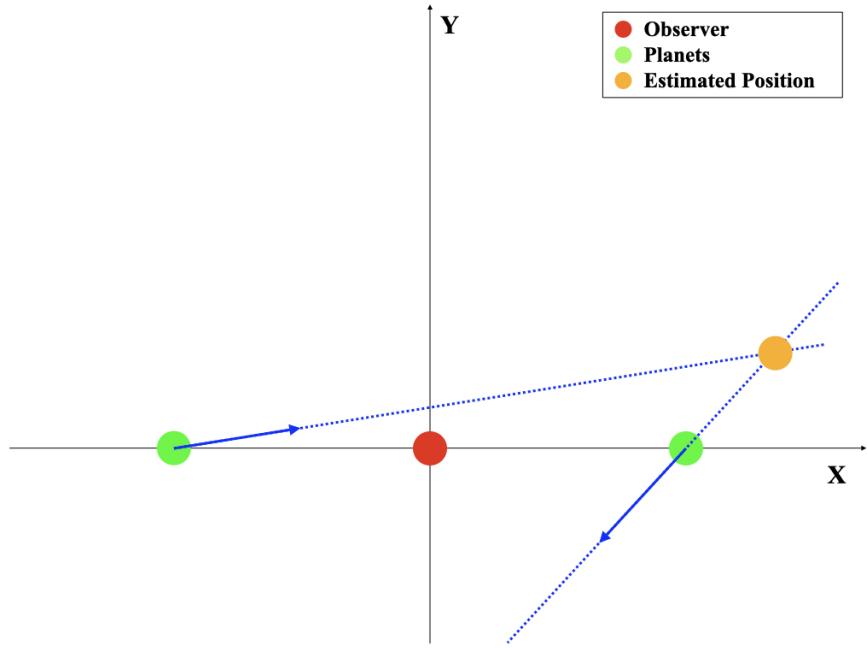


Fig. 16 치명적 오차 발생 경위

2. <Case #2 - 2개의 행성 관측, 치명적 오차 관찰>

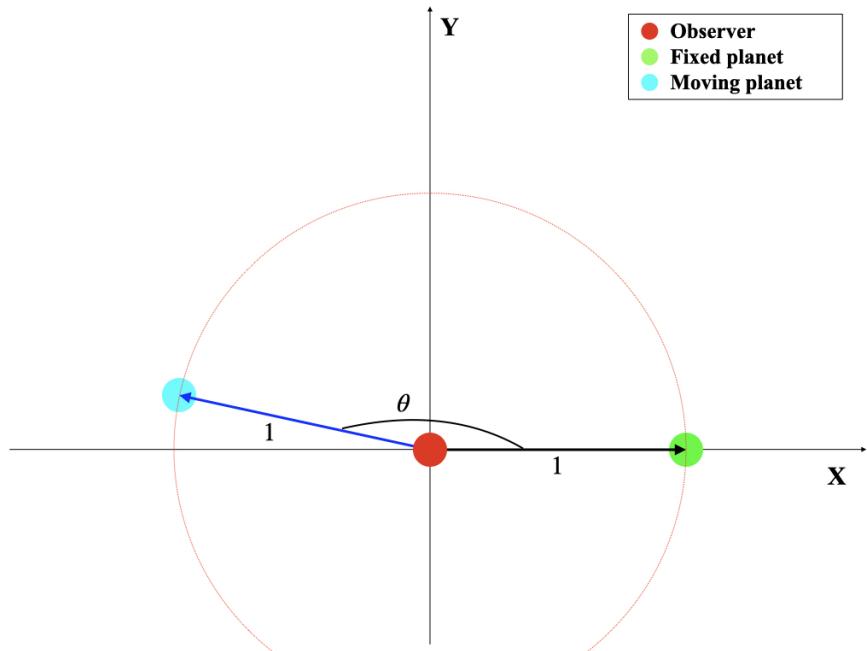


Fig. 17 Case #2

앞서 정의한 치명적 오차의 발생빈도를 확인하기 위해 Case 1에서 오차의 수준의 급격하게 증가한 구간 $0^\circ < \theta \leq 10^\circ$, $170^\circ < \theta \leq 180^\circ$ 에서 치명적 오차의 빈도를 관찰하는 시뮬레이션을 진행한다. 이 때, 관측자-행성 거리는 모두 1로 고정한다. 이 때, 관측자를 중심으로하고 반지름 1인 원 바깥에 위치추정치(Fig. 17의 빨간 점선으로 표현된

원)가 생성되었을 때 치명적 오차가 발생한 것으로 한다.

치명적 오차의 빈도는 행성-관측자-행성이 이루는 각이 일직선($180^\circ, 0^\circ$)에서 멀어질수록 감소한다. 특히, 치명적 오차의 빈도를 0.01% 이하로 줄이려면, 즉, 99.99%로 위치추정치를 신뢰할 수 있으려면 행성-관측자-행성 사이의 각도가 $10^\circ < \theta < 170^\circ$ 인 것이 바람직하다. 이 구간을 벗어나는 $\theta \leq 10^\circ$, $\theta \geq 170^\circ$ 에 해당하는 구간은 앞으로는 치명적 오차가 유의미하게 발생하는 구간이다. 추후 시뮬레이션에서 관측할 수 있는 행성의 숫자가 늘어날 때, 치명적 오차의 빈도가 어떻게 조정되는지, 또 항법의 성능에 어떤 영향을 미치는지 관찰해볼 것이다.

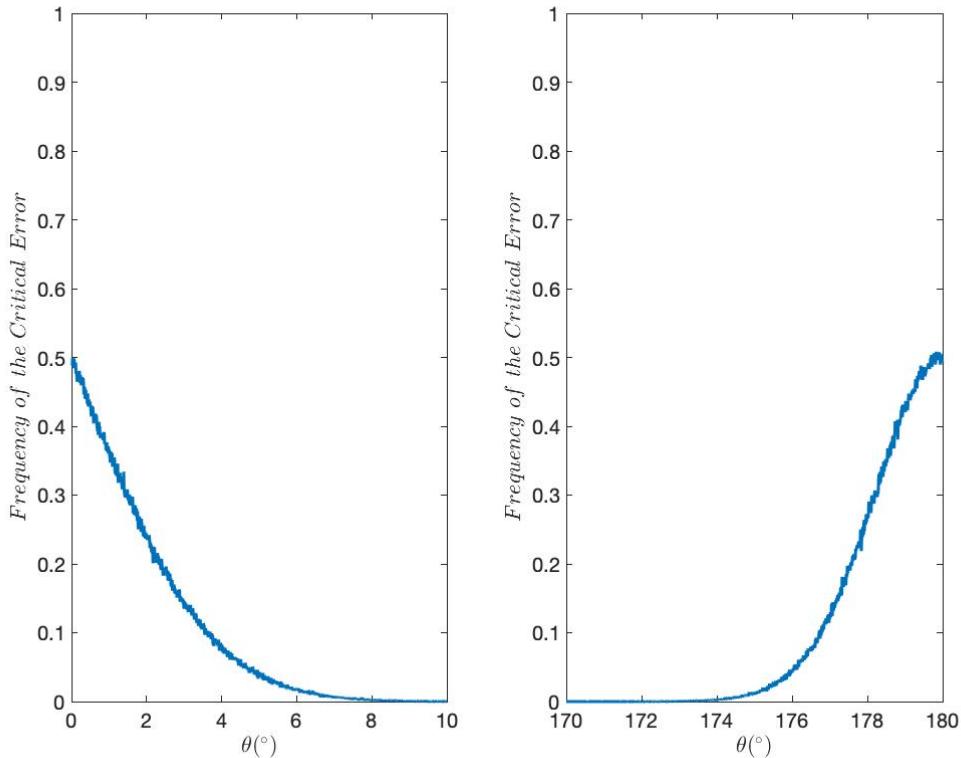


Fig. 18 Case #2의 결과 : θ vs 치명적 오차 빈도 그래프

Table 1 Case #2 결과 : 치명적 오차 빈도

치명적 오차의 빈도(%)	θ 의 범위($^{\circ}$)
< 10	$3.60 < \theta < 176.75$
< 1	$6.61 < \theta < 174.87$
< .1	$8.65 < \theta < 173.39$
< .01	$9.86 < \theta < 170.15$

3. <Case #3 - 3개의 행성 관측>

Case #1, #2 에서는 2개의 행성이 관측된 상황에서 위치추정치를 관찰하였다. Case #3, #4 에서는 3개의 행성이 관측된 상황에서 2-Planet(2개의 행성만 이용할 때) / LS / WLS 세 가지 방법을 모두 이용하여 위치추정치를 생성한다. 앞서 기하학적 이해를 바탕으로 시뮬레이션의 목적으로 삼았던 '3개 이상의 행성을 사용할 때, 위치추정벡터의 조정과정이 알고리즘의 성능에 미치는 영향'을 관찰하기 위해 LS, WLS를 2-Planet과 비교해볼 것이다. 또, 가중치의 영향을 관찰하기 위해 LS와 WLS를 비교해볼 것이다.

Case #3 에서는 Fig. 19과 같이 관측자를 원점에, 2개의 행성을 Case #1에서 가장 좋은 성능을 보여줬던 위치인 $(1, 0)$, $(0, 1)$ 에 각각 배치하고, 세 번째 관측된 행성의 위치를 바꾸어가면서 위치추정치를 관찰한다.

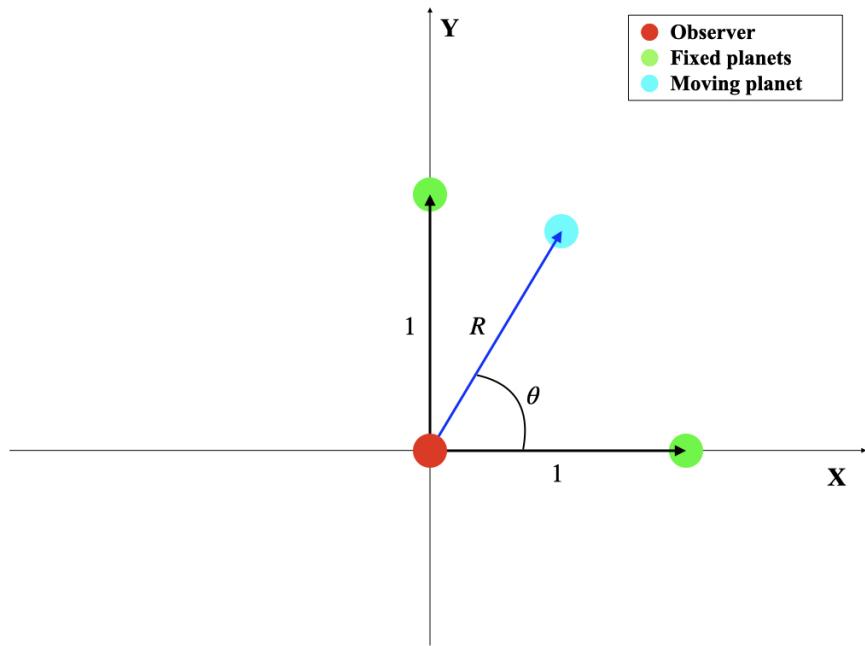


Fig. 19 Case #3

결과(Fig. 20)를 통해 세 번째 행성의 방향은 오차의 크기에 영향을 주지 않는다는 것을 알 수 있었다. 또한, 세 번째 행성이 나머지 두 행성과 비슷한 수준의 거리에 있을 때에는 LS/WLS 모두 2-Planet 보다 좋은 성능을 보였지만, 거리가 증가함에 따라 WLS는 2-Planet과 같은 수준의 오차를 보였고, LS는 오차가 선형적으로 증가해 오히려 2-Planet보다 성능이 떨어졌다. 따라서 3개 이상의 행성을 이용하여 항법을 한다면 WLS를 사용하는 것이 바람직하며, 가중치가 기대했던 역할을 해주고 있음을 확인할 수 있었다.

정리하면, 관측자-행성 사이의 거리를 생각해봤을 때 WLS의 경우 가장 가까이 있는 행성 2개로 위치추정치를 생성하는 것과 같은 결과를 얻을 수 있다. 따라서 일반적인 상황에서 3개 이상의 행성을 사용해도 LS/WLS 모두 2개 행성을 이용하는 것보다 더 좋은 성능을 기대하기는 어려우며, 오히려 계산량만 증가시킨다. 위치추정치를 시각화한 산점도(Fig. 21)와 결과 표(Table. 2)를 확인해보면, 위의 결과를 시각적, 수치적으로 더 명료하게 이해할 수 있다. 특히, LS의 경우 세 번째 행성이 목적함수에 미치는 영향이 WLS보다 더 커서 위치추정치가 세 번째 행성

방향에 수직한 방향(각오차가 영향을 미치는 방향)으로 더 넓게 분포되어 있는 것을 확인할 수 있다.

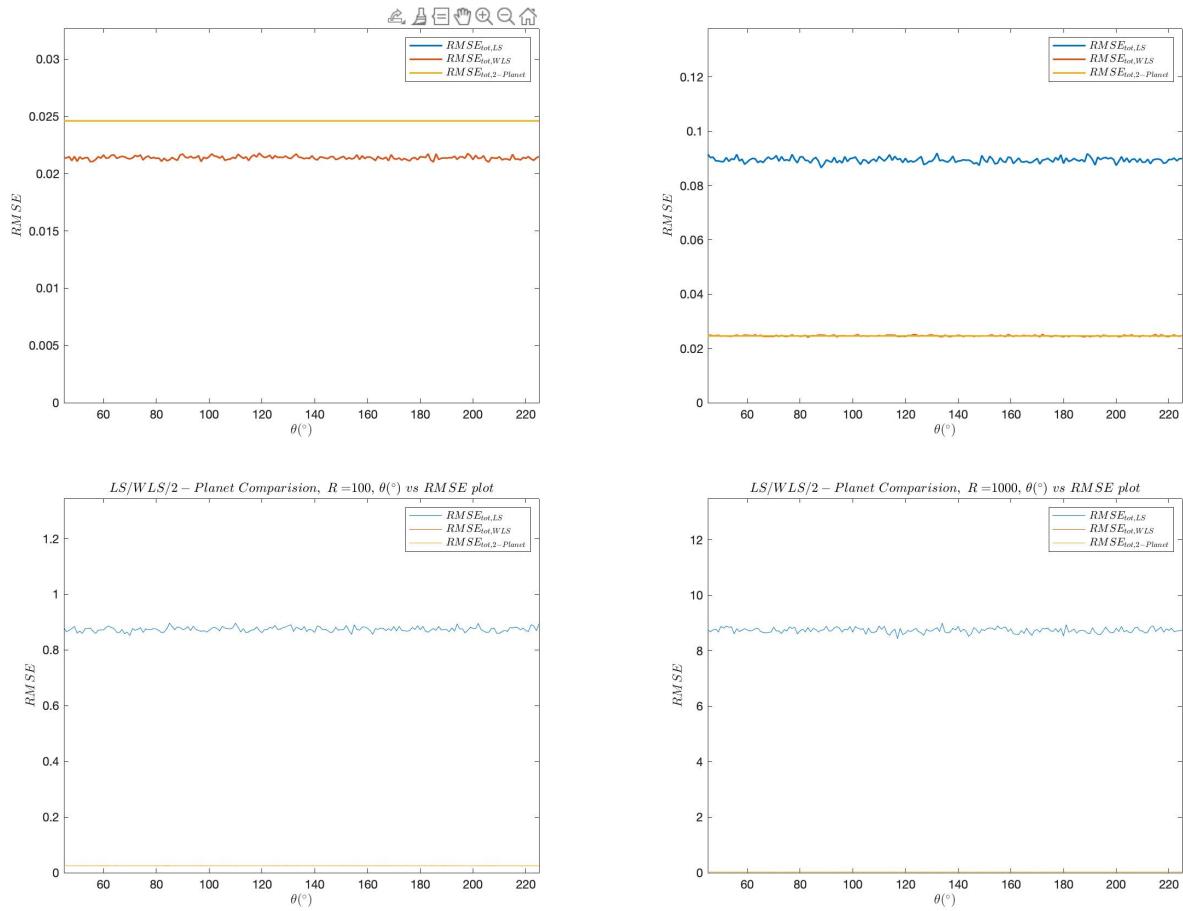


Fig. 20 Case #3의 결과 : θ vs RMSE, 왼쪽 위부터 차례대로 $R = 1, 10, 100, 1000$

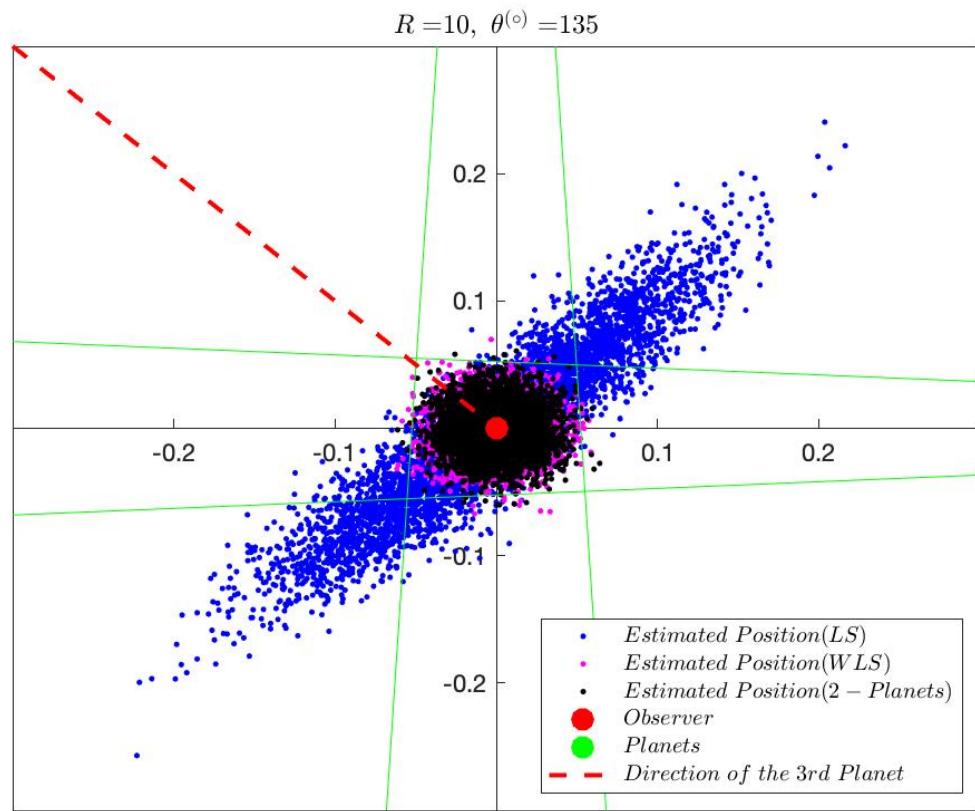


Fig. 21 Case #3의 결과 : $R = 100, \theta = 135^\circ$ 일 때, 위치추정치 시각화

Table 2 Case #3 결과 : $R = 100, \theta = 135^\circ$ 일 때, RMSE

	RMSE
2-Planet	0.0246
LS	0.0890
WLS	0.0248

4. <Case #4 - 3개의 행성 관측, 치명적 오차 관찰>

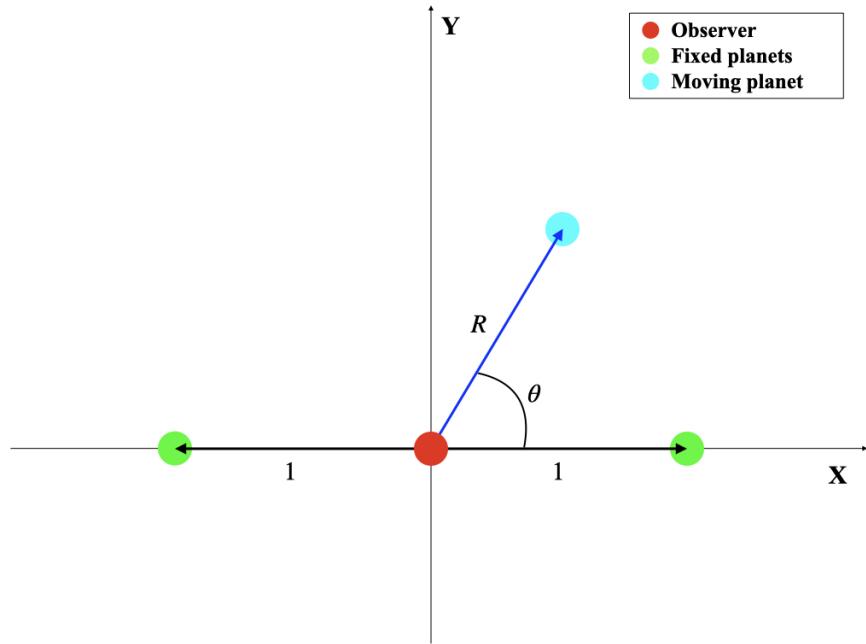


Fig. 22 Case #4

Case #2 에서는 두 행성의 위치변화에 대한 치명적 오차의 빈도를 관찰했다. Case #4 에서는 2개의 행성과 관측자가 정확히 직선을 이루며 배열되어있을 때(가장 치명적 오차의 빈도가 클 때), 세 번째 행성의 거리와 방향을 바꾸어가며 치명적 오차의 빈도와 오차의 크기를 관찰한다.

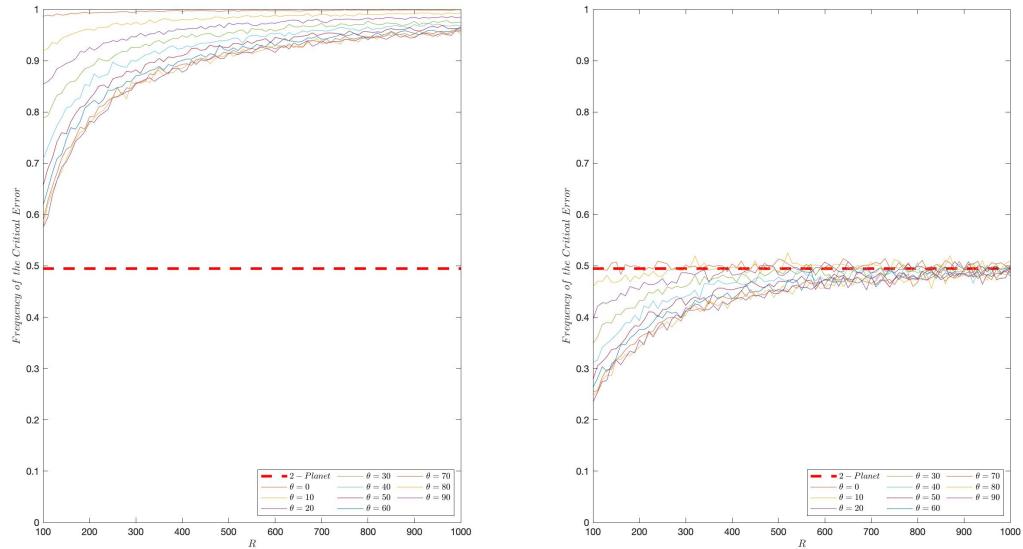


Fig. 23 Case #4의 결과 : R, θ 에 대한 LS(왼쪽)/WLS(오른쪽)의 치명적 오차 빈도 그래프

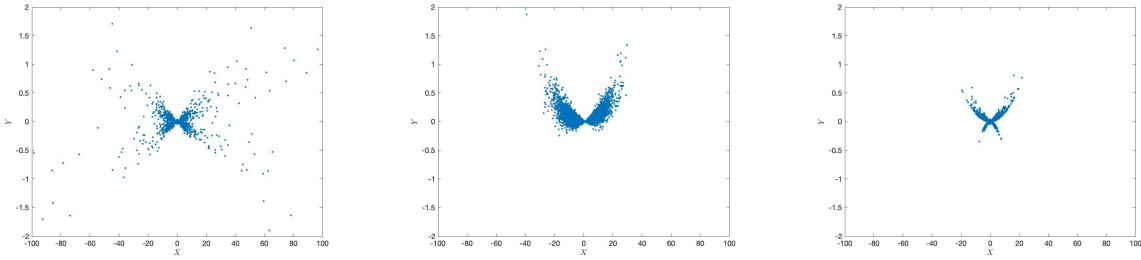


Fig. 24 Case #4의 결과 : $R = 500, \theta = 90^\circ$ 일 때, 2-Planet(왼쪽)/LS(가운데)/WLS(오른쪽) 위치추정치 시각화

Table 3 Case #4 결과 : $R = 500, \theta = 90^\circ$ 일 때, RMSE

	RMSE
2-Planet	132.7010
LS	8.8434
WLS	2.6857

지구-달 거리와 금성-화성 거리를 고려했을 때, LS는 오히려 치명적 오차의 빈도를 높이는 결과 보여준다. WLS는 방향과 거리에 따라 50% 이내로 치명적 오차의 빈도를 줄일 수 있다. 하지만 LS/WLS 모두 위치추정치의 분포가 2-Planet보다 더 밀집되어있는 것을 확인할 수 있다(Fig. 24, Table 3). 그 결과 두 알고리즘 모두 2-Planet에 비해 오차수준이 확연하게 줄어드는 것을 확인할 수 있다. 따라서, 제일 가까운 두 행성과 관측자가 치명적 오차가 유의미하게 발생하도록 놓여있는 상황에서는 2-Planet보다 LS, WLS를 사용하는 것이 바람직하며, 특히 WLS는 치명적 오차의 빈도와 오차수준을 모두 감소시킨다.

B. 시뮬레이션 2 - 달 탐사궤도

앞선 시뮬레이션1에서는 행성들을 임의로 배치하여 거리, 각도가 각 항법알고리즘의 성능에 미치는 영향을 도출했다. 또한, 치명적 오차의 개념을 도입하여 오차가 급격히 증가할때의 행성들의 배열을 찾아내고, 이 배열상에서 각 항법알고리즘의 성능의 차이를 비교했다. 관찰한 바를 바탕으로 실제 달 탐사 궤도 위에서도 시뮬레이션1에서의 결과가 유효한지 판단하기 위해 각 항법알고리즘의 성능을 비교 및 검증한다. 공통사항은 다음과 같다.

- 2021년 1월 1일 00시 00분에 타원궤도 진입을 시작으로 405890초 이후 2차 추진 전까지 호만전이궤도()에서 시뮬레이션을 진행한다[9].
- 달 궤도는 반지름으로 하는 원궤도로 가정하여 진행한다.
- 단순화를 위해 2D 평면(달 궤도면)위에서 진행하며, 금성, 화성의 좌표는 DE405를 이용해 구한 좌표를 달 궤도면에 정사영한 값으로 진행한다.
- 관측자-행성 방향벡터의 각오자는 표준편차가 $10''$ 이고, 평균이 0인 표준편차로 가정하여 진행한다.
- 2-Planet(지구, 달)/LS/WLS 모두 같은 측정치를 이용하여 위치추정치를 생성한다.

1. <Case #5 - 특정 위치>

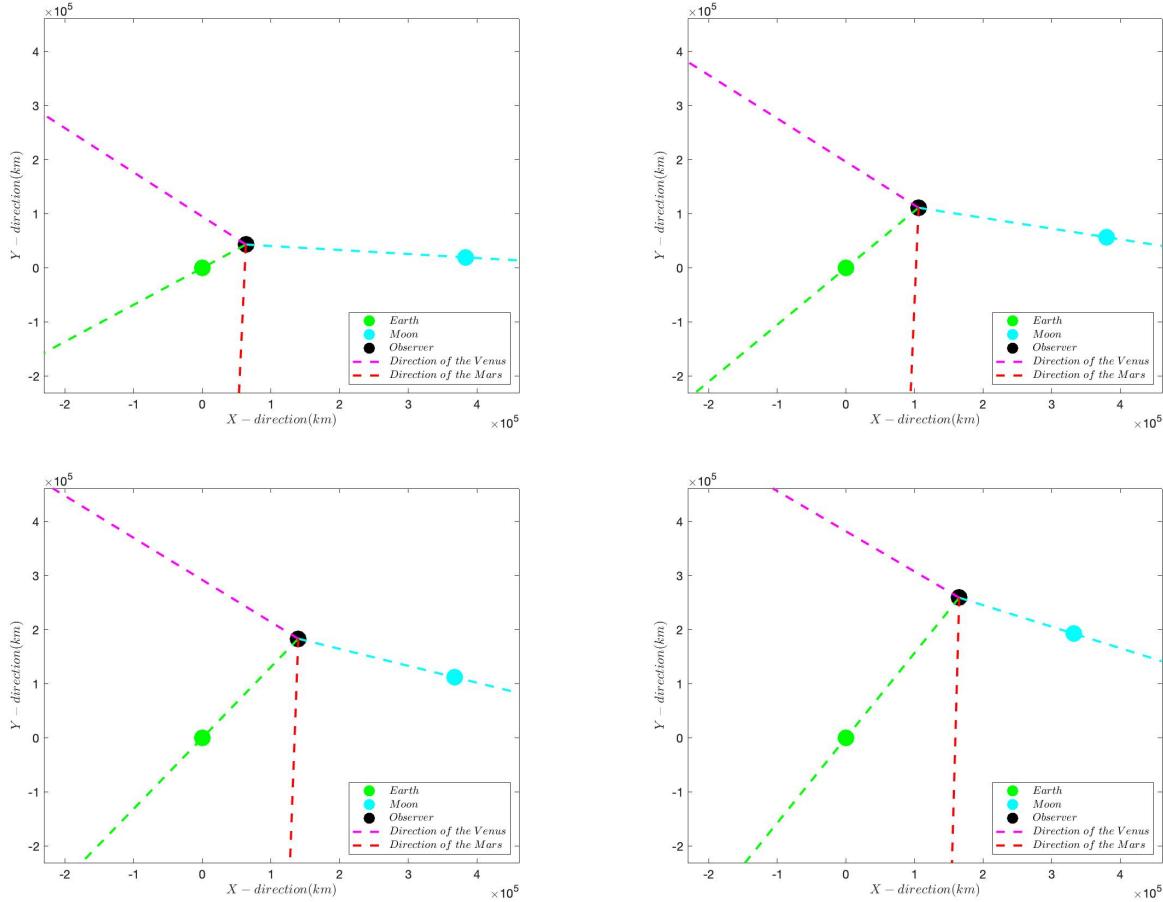


Fig. 25 Case #5 : 원쪽 위에서부터 관측자-지구 사이 거리가 $0.2R, 0.4R, 0.6R, 0.8R$ 일 때 위치관계

Table 4 Case #5 : 각 지점에서 행성-관측자 사이 거리와 지구-관측자-달 사잇각

	$0.2R$	$0.4R$	$0.6R$	$0.8R$
지구-관측자 거리(km)	0.7688×10^5	1.5376×10^5	2.3064×10^5	3.0752×10^5
달-관측자 거리(km)	3.2122×10^5	2.7948×10^5	2.3868×10^5	1.7990×10^5
금성-관측자 거리(km)	2.3335×10^8	2.3356×10^8	2.3389×10^8	2.3438×10^8
화성-관측자 거리(km)	1.3391×10^8	1.3455×10^8	1.3549×10^8	1.3691×10^8
지구-관측자-달 사잇각($^\circ$)	141.53	122.37	109.97	100.85

먼저 특정 위치에서 시뮬레이션을 진행한다. 지구중심-달 사이의 거리를 $R(= 384,400\text{km})$ 라고 했을 때, 호만전 이궤도 위에서 지구-관측자 사이의 거리가 $0.2R, 0.4R, 0.6R, 0.8R$ 인 위치에서 50000번의 몬테카를로 시뮬레이션을 진행한다. 이 때, 지구-관측자-달 사이의 각도는 각각 $141.53^\circ, 122.37^\circ, 109.97^\circ, 100.85^\circ$ 이며, 모두 치명적 오차가 유의미하게 발생하는 구간에 포함되지 않는다. 따라서 행성들 사이의 각도와 거리 수준에 따라 항법알고리즘의 성능이 결정될 것임을 예측할 수 있다. 또한, 금성-관측자, 화성-관측자의 거리수준이 지구-관측자, 달-관측자

거리수준의 1000배 가량이므로, LS의 오차가 2-Planet/WLS의 오차보다 1000배가량 클 것임을 예측할 수 있다.

오차 산점도(Fig. 26, Fig. 27)를 통해 앞서 예측한 바가 옳음을 확인할 수 있다. 또한, 2-Planet/WLS의 경우 가까이 있는 두 행성(지구와 달, 푸른색 계열 점선)에 의해 산점도의 모양이 결정되며, LS의 경우 더 멀리 있는 두 행성(금성과 화성, 붉은색 계열 점선)에 의해 산점도의 모양이 결정되는 것을 확인할 수 있다. 특히, 2-Planet/WLS의 경우 지구와 더 가까울 때에는($0.2R, 0.4R$) 달의 방향벡터 측정치 오차가 목적함수에 미치는 영향이 더 커져 달 방향(Fig. 27의 시안색 점선)에 수직한 방향으로 오차가 더 넓게 퍼져있고, 달과 더 가까울 때에는($0.6R, 0.8R$) 지구의 방향벡터 측정치 오차가 목적함수에 미치는 영향이 더 커져 지구 방향(Fig. 27의 연두색 점선)에 수직한 방향으로 오차가 더 넓게 퍼져있는 것을 확인할 수 있다. LS의 경우 목적함수에 가장 많은 영향을 미치는 금성과 화성의 위치에 의해 전반적인 오차의 크기와 분포가 결정되며, 특히 금성-관측자의 거리가 화성-관측자의 거리보다 약 1.8 배정도 크기때문에, 위치추정오차가 전반적으로 금성 방향(Fig. 26의 마젠타색 점선)에 수직한 방향으로 더 넓게 퍼져있는 것을 확인할 수 있다. 이것들은 앞서 진행한 시뮬레이션1의 결과를 통해 예측할 수 있는 바와 일치한다.

Table 5 Case #5의 결과 : 각 지점에서의 RMSE(km)

	0.2R	0.4R	0.6R	0.8R
2-Planet	25.7124	18.3387	17.1643	17.5970
LS	6.6460×10^3	6.2697×10^3	6.0860×10^3	6.0938×10^3
WLS	25.7122	18.3387	17.1642	17.5969

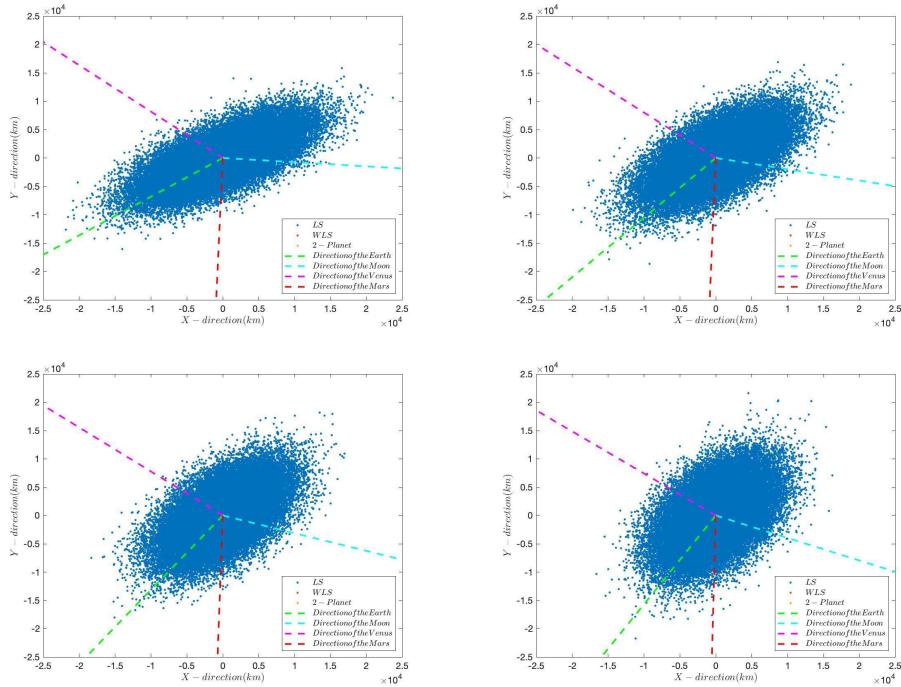


Fig. 26 Case #5 : 원쪽 위에서부터 관측자-지구 사이 거리가 $0.2R, 0.4R, 0.6R, 0.8R$ 일 때 2-Planet/LS/WLS 위치 추정오차 산점도

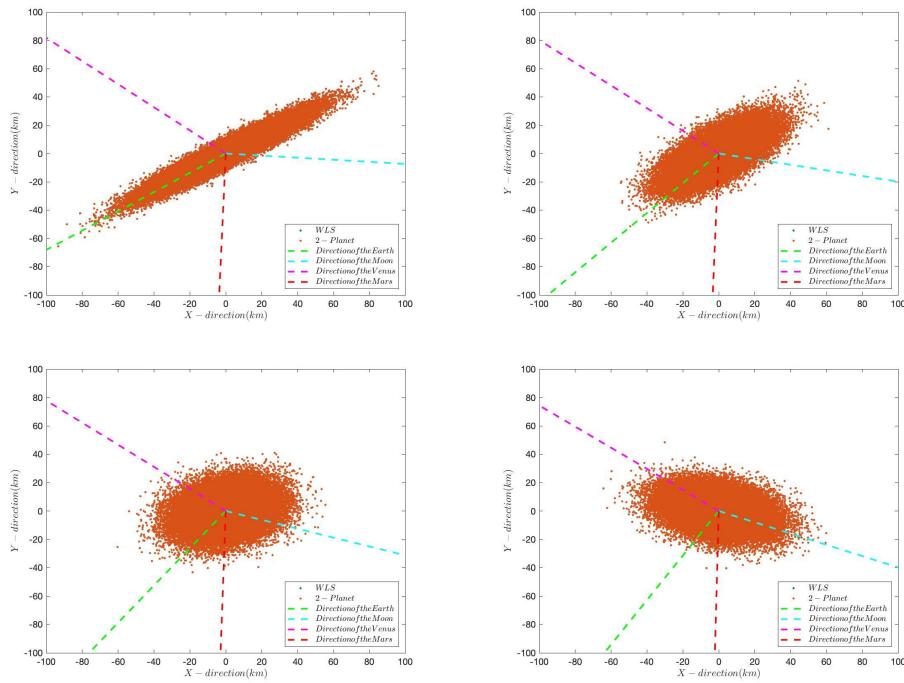


Fig. 27 Case #5 : 원쪽 위에서부터 관측자-지구 사이 거리가 $0.2R, 0.4R, 0.6R, 0.8R$ 일 때 2-Planet/WLS 위치추정오차 산점도

2. <Case #6 - 전체 궤도>

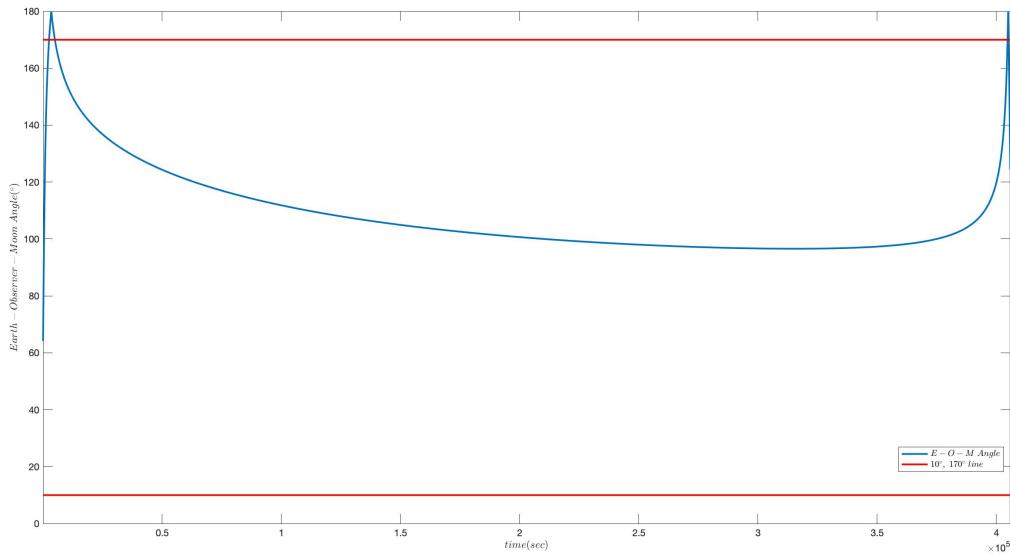


Fig. 28 Case #6 : 시간에 대한 지구-관측자-달 사잇각 변화

호반전이궤도 위에서 1초마다 위치추정치를 생성하여 시뮬레이션을 진행한다. 관측자와 가장 가깝게 위치하는

행성인 지구, 달과 관측자가 이루는 각을 계산하고, Case #2에서 얻어낸 치명적 오차가 유의미하게 발생하는 구간 안에 있는 시간범위에 유의하여 위치추정오차를 관찰한다. Fig. 28에 그려진 빨간 실선은 치명적 오차가 유의미하게 발생하는 경계를 나타낸 것이다. 또한, WLS의 가중치를 이전시간의 위치추정치를 기반으로 정의했으므로, WLS 방법을 이용할 때 가장 처음에 생성하는 위치추정치는 LS로 구하였다.

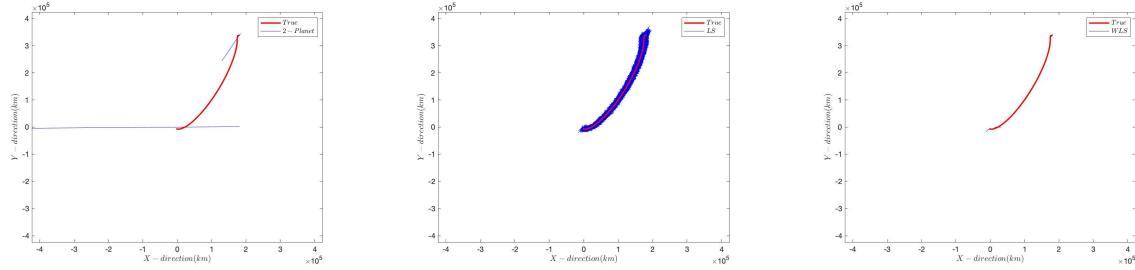


Fig. 29 Case #6의 결과 : 2-Planet(왼쪽)/LS(가운데)/WLS(오른쪽) 위치추정치 시작화

Table 6 Case #6의 결과 : 각 알고리즘의 구간별 RMSE(km)

	2-Planet	LS	WLS
전체시간구간	1.3313×10^3	6.2395×10^3	84.1416
치명적 오차가 유의미하게 발생하는 구간을 제외한 시간구간	20.1102	6.2697×10^3	30.2331(20.1104)

Table 5은 전체시간구간, 치명적 오차가 유의미하게 발생하는 구간을 제외한 시간구간에 대한 각 항법알고리즘의 RMSE값을 나타낸 것이다. WLS 괄호 안에 적혀있는 값은 가장 처음 위치추정치를 생성한 시간을 제외한 구간에서의 RMSE 값이다. LS의 경우, 구간에 관계 없이 다른 항법알고리즘에 비해 높은 수준의 오차를 보인다. 이는 금성-관측자, 화성-관측자 사이의 거리와 지구-관측자, 달-관측자 사이의 거리의 차이로 인해 발생하는 것으로 보인다. 2-Planet의 경우 치명적 오차가 유의미하게 발생하는 구간을 제외한 구간에서는 100km이내의 양호한 오차수준을 보여주나, 치명적 오차가 유의미하게 발생하는 구간에서 발생한 오차로 인해 전체구간에서의 오차수준이 100배 가량 증가하였다. 하지만 WLS는 치명적 오차의 빈도와 그 크기를 줄여 전체 구간에서 100km이하의 안정적인 수준의 오차를 보여준다. 이는 앞에서 진행했던 시뮬레이션1을 통해 예측할 수 있는 바와 일치한다.

V. 결론

위 연구를 통해 행성의 궤도력, 별 추적기와 WLS 알고리즘을 활용하여 달 탐사 궤도 위에서 항법을 할 수 있는 방법을 제시하고, 단순화를 통해 그 성능을 검증했다. 그 과정에서 성능에 가장 큰 영향을 미치는 것이 관측자와 가장 가까운 거리에 있는 두 행성(지구, 달)의 관측자와의 상대적인 위치임을 알 수 있었다. 특히 오차가 급격히 발생하는 상황(지구-관측자-달이 일직선에 가깝게 놓이는 상황)에서 WLS 알고리즘이 갖는 강점을 확인했으며, 그 오차수준이 전반적으로 100km이내임을 확인할 수 있었다. 임무 목표궤도 설정 단계에서 궤도를 적절히 설계하여 탐사선이 지구, 달과 일직선상에 놓이지 않게 한다면 위의 알고리즘을 더욱 안정적으로 운용할 수 있을 것이다. 또한, 추가 연구를 통해 행성-탐사선간의 역학적 모델을 이용하여 필터 이론을 접목하거나, INS, GNSS(지구 근궤도 운용

시) 등 다른 항법 시스템을 서로 보강하는 방식으로 운용할 수 있다면 더 정확하고 정밀한 항법 시스템을 구축할 수 있을 것이다.

References

- [1] Liebe, C. C., “Accuracy performance of star trackers-a tutorial,” *IEEE Transactions on aerospace and electronic systems*, Vol. 38, No. 2, 2002, pp. 587–599.
- [2] Liebe, C. C., “Star trackers for attitude determination,” *IEEE Aerospace and Electronic Systems Magazine*, Vol. 10, No. 6, 1995, pp. 10–16.
- [3] 주광혁, and 이상률, “별추적기의 기술개요와 개발동향,” *한국항공우주학회지*, Vol. 38, No. 3, 2010, pp. 300–308.
- [4] Conway, D., and Mortari, D., “Single-Point and filtered relative position estimation for visual docking,” *International Conference on Computational & Experimental Engineering and Sciences (ICCES), Reno, NV*, 2015.
- [5] Mortari, D., and Conway, D., “Single-point position estimation in interplanetary trajectories using star trackers,” *Celestial Mechanics and Dynamical Astronomy*, Vol. 128, No. 1, 2017, pp. 115–130.
- [6] Williams, D. R., “JPL Planetary and Lunar Ephemerides,” , 2019. URL https://ssd.jpl.nasa.gov/?planet_eph_export/.
- [7] Petit, G., and Luzum, B., “IERS conventions (2010),” Tech. rep., Bureau International des Poids et mesures sevres (france), 2010.
- [8] Park, R., “Planetary Fact Sheet - Metric,” , 2020. URL <https://nssdc.gsfc.nasa.gov/planetary/factsheet/>.
- [9] 강민석, “한국형 달탐사의 항법을 위한 궤도 설계,” 서울대학교 항공우주공학과 졸업논문, 2020.