

*De  
revolutionibus orbium coelestium  
et  
momentis vehiculorum liberorum*

*auctore*

*Alberto Liberio Humaniense Disciplino*

*die*

*vicesima Augusta MMXVI*

*factum magnum super regnum publicam  
imperii super libertatem et veritatem  
democratia proficit in fascismum et nazismum  
libertas proficit in mendacium publicam  
sed verum dictum a te aliquando reveletur  
rerum litterae aeternum permaneant  
possit nemo prohibere*

*feb mmxvi alhd*

## Kerbal Space Program이란?

Kerbal Space Program (약칭 KSP) 란 Squad사에서 만든 우주선 시뮬레이션 게임입니다. 간단히 말하자면, 이 게임은 뉴턴역학을 충실히 구현해 로켓역학을 배울 수 있는, 체험할 수 있는 게임입니다. 다만, 좀 더 하드코어 시뮬레이션이지만 (실제 아폴로 미션을 구현하는 등) 한정된 상황만을 체험해 볼 수 있는 다른 우주선 게임과는 다르게, 샌드박스적 요소가 강한 게임이라고 할 수 있겠습니다.

이 게임에서는 뉴턴 제2법칙 등과 같은 뉴턴 역학을 단순히 교과서에서 배우는 것만으로는 알기 힘든 여러가지 물리적 현상들을 직접적 피드백을 통해 손수 익힐 수 있습니다. 게임 맵 상의 궤도 위에서 로켓분사(매뉴버) 계획을 하면, 얼마만큼 궤도가 변하는지 실시간으로 보여 주므로, 뉴턴역학을 전혀 모른다고 하더라도 직접 만져보면서 흥미로운 로켓공학의 현상들을 익힐 수 있고 미션을 성공시켜 볼 수 있습니다. (처음 궤도에 올리는 데에는 이런 느긋한 방법이 통하지 않을지도 모릅니다. )

이 게임은 그저 실제에 충실히 뉴턴역학을 구현하는 데에만 신경을 쓴 것은 아닙니다. 실제 태양계보다 축소된 모델로 우주선을 궤도에 올리는 것을 실제보다 용이하게 해놓은 것이 많은 사용자들이 더 쉽게 플레이 해볼 수 있게 하는 장점을 가지고 있습니다. 게임에 등장하는 우주비행사도 인간이 아닌 작고 귀여운 녹색인간으로, 캐릭터성 또한 겸비하고 있습니다. 이와 같은 쉬운 게임 전개는 다음과 같은 코멘트가 많은 추천을 받게 하기도 했습니다: 딸도 몇 번의 시행착오를 거쳐서 달까지 우주선을 보낼 수 있었다.

이처럼 학생들에게 물리를 배울 동기를 제공하거나 로켓공학의 기초지식을 전달하는 데에 효과가 있고, 제작사에서도 교육용을 따로 판매하고 있을 정도로 교육용도로서의 홍보에 적극 적입니다. 학교에서 단체로 구매한 다음, 학생들에게 숙제를 내어줄 수도 있습니다. 또한, 2차 창작에도 유연하여, 링크만 건다면 게임 영상을 유튜브에 올릴 수 있도록 홈페이지에서 명시적으로 허락하고 있습니다. 많은 이용자들이 자신만의 우주선이나 비행 방법을 올리고 있고, 어떤 사람은 멋진 우주 영화를 만든 사람도 있습니다. 이처럼 다른 게임에 비해 오픈된 마인드와 샌드박스이면서도 하드코어 하지 않은 게임성 덕분에 더욱더 많은 이용자들이 유입되고 있고 앞으로의 기능확장에도 무궁무진한 가능성의 열려 있는 게임이라고 할 수 있습니다.

하지만, 새롭게 이 게임을 시작하는 사람들이 진행에 많은 어려움을 겪는 것을 보았습니다. 처음 궤도에 우주선을 올릴 때에는 단순한 관성계의 뉴턴역학만으로는 성공시킬 수 없기 때문입니다. 첫 발사를 성공시키기 위해서는 공기역학을 이해해야 하고, 게임상의 엔진의 성질에 대해서도 이해할 필요가 있습니다.

이 서적의 목적은 크게 두 가지입니다. 우선, 처음 입문하는 사람들이 쉽게 따라하면서 게임을 이해할 수 있는 입문서로서의 목적이 있습니다. 또한, 가능한한 많은 수치표를 넣어 숙련자들도 이용할 수 있는 계산툴로서 기능하게 하는 목적이 있습니다. 최대한 첫째 목적을 이루는 것이 주 목표이며, 이를 지키면서도 최대한 많은 물리현상들을 설명하고, 또 정교한 스킬들을 익힐수 있게 하고자 합니다. 짧은 집필기간과 계속되는 게임의 업데이트 때문에 초판은 충분히 잘 기술되지 못할 수 있을 것입니다. 게임소스를 제외한 본 서적의 핵심적인 부분인 기술적인 설명과 저자의 개인연구 부분은 오픈소스 라이센스 하에 있으며, 앞으로 독자들께서도 자유롭게 편집할 수 있도록 할 것이오니, 혹시 틀린 점을 발견하거나 추가하고 싶은 내용이 있으면 책 마지막 장의 소스 URL에서 추가해주시길 부탁드립니다.

# Contents

Kerbal Space Program이란?	3
<b>Part 1. 시작하기</b>	7
Chapter 1. 궤도에 올리기	9
1.1. 운동에너지 얻기	9
1.2. 공기역학 및 안정성	9
1.3. 재진입시 주의사항	10
Chapter 2. 다단 로켓과 $\Delta v$	11
2.1. 연료	11
2.2. 다단 로켓 (Linear Staging)	11
2.3. 아스파라거스 (Asparagus Staging)	12
2.4. $I_{sp}$	12
Chapter 3. 궤도운동	13
3.1. 용어설명 및 주의사항	13
3.2. 원궤도	13
3.3. 타원궤도	13
3.4. 쌍곡선궤도 - 탄성충돌	14
3.5. 슬링샷	14
Chapter 4. 달 미션	15
<b>Part 2. 매뉴버</b>	17
<b>Part 3. 미세조정 (Fine-tuning)</b>	19
<b>Part 4. 행성간 비행</b>	21
Chapter 5. 시간계획 정하기	23

Chapter 6. 포획 (Capturing)	25
6.1. 대기를 이용한 포획 (Atmospheric Capture)	25
6.2. 위성을 이용한 포획 (Sling-shot Capture)	25
6.3. 포획을 위한 궤도조정 (Fine-tuning with Sling-shot Capture)	26
Chapter 7. 사례들	27
7.1. Juno's Maneuver	27
<b>Part 5. 그 외 다루어지지 않은 것들</b>	<b>29</b>
Chapter 8. Aviation	31
Chapter 9. 3체 문제 (Three Body Problem)	33
9.1. KSP의 간략화	33
9.2. 근일점 변화	33
9.3. 세차운동	33
Chapter 10. 상대론적 문제	35
10.1. 특수상대론: 로렌츠변환	35
10.2. 특수상대론: 적색/청색편이	35
10.3. 일반상대론적 효과 및 천체모델의 근사성	35
<b>Part 6. 데이터 및 프로토콜</b>	<b>37</b>
Chapter 11. 게임 데이터	39
11.1. 행성 데이터	39
11.2. 위성 데이터	40
11.3. 엔진 데이터	40
Chapter 12. 프로토콜	41
12.1. Kerbin 궤도 프로토콜	41

## Part 1

# 시작하기



## CHAPTER 1

# 궤도에 올리기

### 1.1. 운동에너지 얻기

**일의 공식**      일의 공식은  $W = \vec{F} \cdot \vec{v} = Fv \cos \theta$ 이다. 즉, 힘의 방향이 운동방향과 나란할 수록 얻는 운동에너지가 많아진다. 따라서 로켓을 발사할 때는 속도 방향과 최대한 가까운 방향으로 가속해주는것이 가속효율에 도움이 된다. 단, 에너지는 기준계에 따라 달라지므로 어떤 기준계에서 생각하는지가 중요하다. 어떤 행성 주위에 궤도를 형성하고 싶다면 그 행성을 기준으로 한 좌표계에서 생각해야 한다. KSP에서는 어떤 행성이나 위성 가까이에 있으면 그 별을 중심으로 하는 관성계를 기준으로 삼고 있다 (comoving inertial frame). 태양이나 모행성에 의한 기조력은 생각하지 않고 있으므로 훨씬 간단하게 계산할 수 있지만, 또한 나름 실제와 비슷한 현상을 이와 같은 저차근사에서도 볼 수 있다. 주의해야 할 점은 기준계를 행성표면을 따라 자전하는 좌표계로 잡게 되면 비관성계의 역학을 적용해야 하므로 뉴턴역학을 적용할 수 없다는 점이다. 따라서, 공기역학을 다루지 않고 순전히 탄도학만 생각할 때는 관성계를 항상 기준으로 해야함을 잊지 않아야 한다.

**Gravity turn.**      처음 로켓을 발사할 때는 수직으로 발사하게 된다. 하지만 행성 주변으로 원운동 (혹은 타원운동)을 하게 하기 위해서는 지면과 수평방향의 속도를 주는 것이 중요하다. 이 때, 로켓의 방향을 수직에서 수평으로 눕혀주는 속도가, 앞에서 말한 바와 같이 에너지 효율에도 중요하고, 로켓의 공기역학적 안정성에도 중요하다. 이러한 각도변화는 인위적으로 조정해야 하지만 (pitch manuever) 어느 정도 중력에 의해 자연적으로 일어난다 (gravity turn). 여기서는 이러한 turn이 자동적으로 이루어지는 이유와 정도에 대해서 우선 설명하고, 그 다음 이러한 turn을 얼마나 조정해야 하는지 설명하도록 할 것이다.

### 1.2. 공기역학 및 안정성

#### 1.2.1. Over-compensating.

### 1.3. 재진입시 주의사항

1.3.1. 방열판 (Heat Shield).

1.3.2. 낙하산 (Parashute). | 이름 | 작동속도 | 면적 |

## CHAPTER 2

### 다단 로켓과 $\Delta v$

#### 2.1. 연료

$$(1) \quad \Delta v = I_{sp} g_0 \log \frac{M + m}{M}$$

$$(2) \quad \Delta v = I_{sp} g_0 \log \frac{M + (1 + \alpha)m}{M + \alpha m}$$

$$(3) \quad \Delta v \rightarrow I_{sp} g_0 \log(1 + \alpha^{-1})$$

is independent of thrust force but only depend on isp and etc

Let's assume  $I_{sp} = 320s$  and

Estimated Engine Mass / Fuel Mass = 1/6

Estimated Fuel Tank Mass / Fuel Mass = 1/8

$$\Delta v = 4669.78m/s$$

실제로는 1단의 경우 공기저항으로 인해 연료를 많이 실을 수록 오히려 얻을 수 있는 운동에너지가 줄어드는 결과를 보이기도 한다. 따라서 많은 양의 화물(Load)을 쏘아 올리기 위해서는 '다단 로켓 (multi-stage rocket)'과 '아스파라거스 로켓 (asparagus-staging rocket)'이 필요하다.

#### 2.2. 다단 로켓 (Linear Staging)

a

### 2.3. 아스파라거스 (Asparagus Staging)

#### 2.4. $I_{sp}$

$$(4) \quad I_{sp} = \frac{F}{\dot{m}g_0}$$

## CHAPTER 3

# 궤도운동

### 3.1. 용어설명 및 주의사항

\*근일점은 꼭 태양과 지구사이만의 의미가 아니고 일반적으로 사용할것임

### 3.2. 원궤도

### 3.3. 타원궤도

별 (start) 우주선 (projectile) 위치에너지 (potential energy)

$$(5) \quad r^2\dot{\theta} = l$$

$$(6) \quad \ddot{r} - \frac{l^2}{r^3} + \frac{GM}{r^2} = 0$$

$$(7) \quad -l^2r^{-2}\frac{d^2r^{-1}}{d\theta^2} - l^2r^{-3} + GMr^{-2} = 0$$

$$(8) \quad \frac{d^2r^{-1}}{d\theta^2} + r^{-1} - GMl^{-2} = 0$$

이러한 우주선(projectile)의 운동방정식의 해는 다음과 같다.

$$(9) \quad r^{-1} = \frac{GM}{l^2} + \sqrt{\frac{2\epsilon}{l^2} + \left(\frac{GM}{l^2}\right)^2} \cos(\theta + \theta_0)$$

여기서  $\epsilon$ 은 우주선(projectile)의 질량당 총 에너지이다. 이러한 식은 이차곡선(원, 타원, 포물선, 쌍곡선)을 나타내는 표현이다. 따라서 이는 역제곱힘에서의 궤도가 이차곡선이 된다는 증명이다. 이 식을  $l$ 과  $\epsilon$ 이 아닌  $r$ 과 근일점( $r_p$ )의 함수로 나타내면 다음과 같이 나타낼 수도 있다.

$$(10) \quad r^{-1} = \frac{r_p^{-1}}{2} \cdot \frac{1}{1 + \epsilon(GM)^{-1}r_p} + \frac{r_p^{-1}}{2} \cdot \frac{1 + 2\epsilon(GM)^{-1}r_p}{1 + \epsilon(GM)^{-1}r_p} \cos(\theta + \theta_0)$$

$r$ 이 무한대로 가지 않고 유한한 영역에서 진동하고 있으면 원이나 타원, 즉 구속궤도(bound orbit)이 되고,  $r$ 이 무한대, 즉  $r^{-1} = 0$ 인 지점이 있으면 비구속궤도(unbounded orbit)가 될 것이다.

모행성	모행성계로 Hoffman Transfer 할 때 행성계 기준에서의 속도	위성	Radius	Lowest Safe Altitude <sup>1</sup>
-----	---	----	--------	-----------------------------------

### 3.4. 쌍곡선궤도 - 탄성충돌

### 3.5. 슬링샷

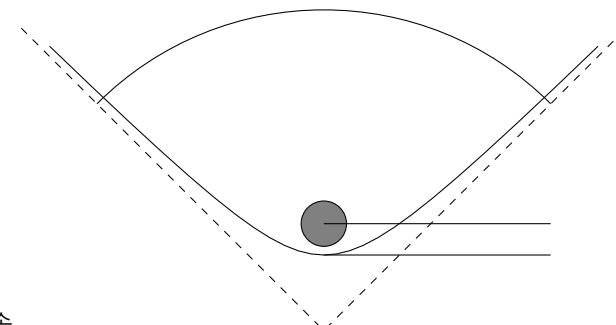
$$(11) \quad \Delta\theta = \pi + 2 \sin^{-1} \frac{1}{1 + 2\epsilon(GM)^{-1}r_p}$$

특akak aaa

#### 충돌각 최대값

**특정 기준계에 대한 슬링샷 결과** 어떤 기준계에 대해서 우주선의 입사 속도가  $\vec{v}_s(t = -\infty)$ , 천체의 속도가 상수  $\vec{v}_c$ 라고 한다면, 위와 같은 계산결과에 따라 우주선의 최종 속도가 어떻게 되는지 계산해보자.

우선 천체계에서 우주선의 속도는  $\vec{v}_s(t = -\infty) - \vec{v}_c$ 가 될 것이다.



예) 달을 이용한 가속

진입속도  $v$ , 근접거리  $r$ .

---

<sup>1</sup>행성의 가장 높은 지점의 고도와 대기의 두께 중 큰 것

## CHAPTER 4

### 달 미션



## Part 2

### 매뉴버



## Part 3

### 미세조정 (Fine-tuning)



Part 4

## 행성간 비행



## CHAPTER 5

# 시간계획 정하기



## CHAPTER 6

# 포획 (Capturing)

행성계 외부에서 진입하는 물체는 탈출속도를 넘어서므로<sup>1</sup> 힘이 작용하지 않으면 쌍곡선 궤도를 그리며 다시 행성계 밖으로 탈출하게 될 것이다. 안정적인 미션 수행을 위해서 행성간 미션에서는 도착지에서 충분히 감속하여 구속궤도를 만들 수 있는 기술이 필요하다. 이를 '포획'이라고 부르도록 하겠다. 포획이 이루어지지 않는다면 행성의 인공위성 궤도에 진입하는 미션을 수행할 수 없으며, 착륙 미션에 경우 한번만에 성공해야 하는 부담을 안게 된다.<sup>2</sup>

엔진을 이용한 능동적인 감속은 설명이 필요없으므로 여기서는 동력을 사용하지 않고 포획당하는 방법에 대해서 설명하고자 한다. 두가지 방법을 생각할 수 있을텐데, 행성의 대기를 이용한 방법과 행성의 위성을 이용한 방법이 있을 것이다. 각각의 방법에 대해서 설명하고자 한다.

### 6.1. 대기를 이용한 포획 (Atmospheric Capture)

### 6.2. 위성을 이용한 포획 (Sling-shot Capture)

다음은 위성을 이용한 슬링샷으로 행성계에서의 속도를 줄일 수 있는지 검토해 보도록 하자. 위성과 조우시 위성의 속도가 우주선의 반대방향이라면 위성의 운동량을 받아 감속할 수 있으리라고 예상할 수 있을 것이다. 하지만 행성간 여행에서 그렇게 정확한 타이밍을 맞출 수 있으리라고 기대하기는 어렵다. 이번 section에서는 행성계 진입후 주어진 상황에서 감속할 수 있는지에 대해 알아볼 것이다. 다음 section에서는 진입 타이밍을 조정하는 법에 대해 논의해 보겠다.

우선 행성과 위성은 충분히 멀리 떨어져 있어서 조우(encounter)를 탄성충돌로 근사할 수 있다고 가정할 것이다. 사실 KSP는 각 천체의 '영향권'내에서 1체문제로 환원함으로서 이러한 가정을 충실히 따르고 있다.

<sup>1</sup>정확히는 '행성계 관점에서 보았을 때 양(陽)의 에너지를 갖으므로'라고 하는 것이 옳을 것이다. '탈출속도'라는 개념은 보통 원궤도를 그리던 물체에 얼마만큼의  $\Delta v$ 가 주어져야 탈출할 수 있는지에 대한 얘기이다.

<sup>2</sup>비구속궤도(쌍곡선궤도)로부터 대기권으로 진입할 때, 높은 진입속력으로 인해 높은 열이 발생하게 되어 우주선이 소실될 위험성이 커지게 되며, 또한 다시 뛰겨나가지 않고 충분히 감속할 수 있는 가능성도 줄어들게 된다.

### 6.3. 포획을 위한 궤도조정 (Fine-tuning with Sling-shot Capture)

## CHAPTER 7

# 사례들

### 7.1. Juno's Maneuver

- (1) 지구계를 탈출한다.
- (2) 약 2배의 공전궤도를 만들어 다시 지구와 만날 수 있게 한다.
- (3) 원일점에서 근일점을 낮게 하는 매뉴버를 실시하여 다음 매뉴버에서의 에너지 효율을 높인다.
- (4) 지구를 이용한 슬링샷 효과를 포함하여 목성과의 접점을 만든다.
- (5) 목성근체에서 목성과 비슷한 속도로 가속하여 목성궤도로 들어간다.



## Part 5

그 외 다루어지지 않은 것들



## CHAPTER 8

# Aviation

이번 chapter에서는 비행기로 비행시에 유용하게 쓸 수 있는 도표를 소개하고자 한다.

비행기의 비행속도 측정시에 IAS(Indicated Airspeed)라는 단위를 쓰게 된다. 이는 비행기 바깥에 붙어있는 압력계를 이용하여 공기밀도를 표준밀도로 가정하고 공기의 유속을 추정한 결과로서, 베르누이 법칙( $\rho v^2 + P = \text{const}$ )을 고려해보자면 계기에 의해 측정되는 수치는 같은 유속일때 유체의 밀도의 (-1/2)승에 비례하게 될 것이다. (반대로 True Airspeed를 구하기 위해서는 측정된 압력차에 공기밀도의 제곱근을 곱하면 될 것이다.)

한편, 공기저항 또는 양력(lift)은 보통 저속에서 면적과 공기밀도에 비례하고 속도의 제곱에 비례한다는 공식을 쓴다. ( $\sim \rho v^2 A$ ) 두 공식이  $\rho v^2$ 에 비례한다는 점에서 같으므로 양력은 IAS에만 의존하게 된다고 할 수 있다. (물론 고속에서 양력 공식이 변한다면 이 또한 달라질 것이다.)

저자가 실제로 게임 상에서 실험해본 바로는, 실제 지구처럼 시각에 따라 태양복사에 따라 대기의 온도가 변하는 듯하지만, 위키에 제시된 Kerbin 표준 기온 도표에 따라 계산하도록 하겠다.<sup>1</sup>

**총단속도** 흡기하는 제트엔진의 추력은 어림잡아 공기밀도에 비례한다고 예상할 수 있다. (실제 게임에서는 어떤 모델을 쓰는지 모르겠음. 추가바람. 특히, 램제트 같은 고속에 특화된 엔진 및 흡기구의 경우 많이 달라질 수 있음.) 또한, 공기저항도 어림잡아 공기밀도에 비례하므로, 비행기의 총단속도는 고도에 따라 변하지 않는다고 예상할 수 있다.

경험상 흡기엔진으로 구동하는 비행기가 항속 가능한 최대고도 및 속도는 23000 m 및 1200m/s 정도였다. (정확한 자료 추가바람. 하지만 비행기의 모델에 따라 달라지므로 정말 '극한값'을 구할 수 있을지는 미지수.) 또한, 흡기 자체는 26000 m 까지 가능하였다. (속도에 따라 달라질 수 있다.)

---

<sup>1</sup><http://wiki.kerbalspaceprogram.com/wiki/Kerbin>

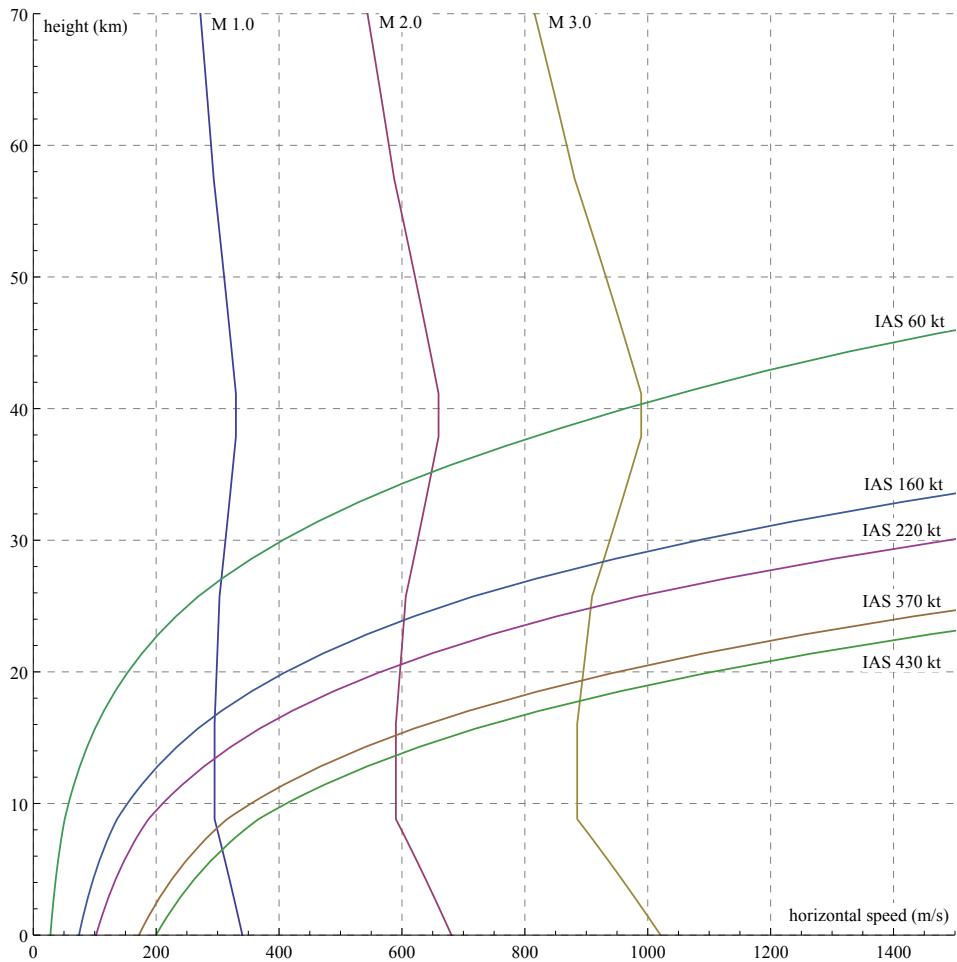


FIGURE 1. test 참고로 대략적으로 IAS 60 kt는 경비행기의 이착륙속도 이고 IAS 160 kt, 220 kt는 각각 B747과 Concorde의 이착륙속도이다. IAS 370 kt와 430 kt는 747과 Concorde의 순항속도이다. 실제 지구에서 Concorde는 55,000 ft에서 M 2.0으로 항행하였다.

## CHAPTER 9

# 3체 문제 (Three Body Problem)

### 9.1. KSP의 간략화

### 9.2. 근일점 변화

$$(12) \quad \frac{d^2 r^{-1}}{d\theta^2} + r^{-1} - GMl^{-2} - l^{-2}r^2V'(r) = 0$$

$$(13) \quad \frac{d^2 r^{-1}}{d\theta^2} + r^{-1} - GMl^{-2} + 2l^{-2}r^3 \left( \frac{|\alpha_-| - |\alpha_+|}{2} + \frac{|\alpha_-| + |\alpha_+|}{2} \cos(\theta - \omega t) \right) = 0$$

### 9.3. 세차운동

\*로켓발사로 인한 변화  
애초에 축은 0도로 고정



## CHAPTER 10

# 상대론적 문제

- 10.1. 특수상대론: 로렌츠변환
- 10.2. 특수상대론: 적색/청색편이
- 10.3. 일반상대론적 효과 및 천체모델의 근사성



## Part 6

# 데이터 및 프로토콜



## CHAPTER 11

# 게임 데이터

이 챕터의 내용은 주로 게임 내부 데이터 및 물리법칙에 대한 것이며, 따라서 저작권은 게임제작사에 있으며, 인용하고 있는 제3자의 저작권은 없는 것으로 해석한다. (독창성 결여) 데이터는 게임 플레이 중 직접 확인할 수 있는 부분이며, 상이한 점이 있다면 github에서 직접 간접할 수 있다.

인용처는 다음과 같다.

- 게임에서 직접 확인한 정보 인용
- 참고:
- 참고: <http://wiki. kerbalspaceprogram.com/>

### 11.1. 행성 데이터

KSP에서 행성과 위성의 운동은 물리법칙의 시뮬레이션에 의한 것이 아니라 고정된 주기를 가진 고정된 운동궤도에 따른 것입니다. 공전주기와 공전궤도는 항성 또는 모행성이 아주 무겁다고 생각하고 만들어진 1체문제의 해이며, 모행성의 위성에 의한 흔들림 등은 전혀 고려되어 있지 않습니다. 

이름	질량×중력상수	원일점	근일점	공전주기 <sup>1</sup>
----	---------	-----	-----	-------------------

(\*괄호 안의 물리량은 다른 양으로부터 유도되는 양입니다.) (표에 주석은 어떻게 달죠)

**1 Solar Day.** Different than mean solar day, a solar day varies depending on the position on the solar orbit. There are mainly two reasons of the variation of the solar day:

- eccentricity of the orbit,
- and axial tilt or obliquity, angle between equatorial plane and orbital plane.

Under these assumptions:

- the Sun is much more massive than a planet so that the orbit is perfectly ellipse,
- and a day is much shorter than an year,

we can obtain the length of a solar day depending on the position.

First, let us obtain the angular velocity on certain position using Kepler's second law. The total area enclosed by the elliptic orbit on the orbital plane is  $\sqrt{r_p r_a} \frac{r_p + r_a}{2} \pi$  where  $r_a$  is the apoapsis, and  $r_p$  is the periapsis radius. Thus, the areal velocity is constantly  $\sqrt{r_p r_a} \frac{r_p + r_a}{2} \frac{\pi}{T}$  where  $T$  is the orbital period. In conclusion, the equation to obtain the relation between solar distance  $r$  and angular velocity  $\omega$  is:

$$(14) \quad \begin{aligned} \frac{r^2 \omega}{2} &= \sqrt{r_p r_a} \frac{r_p + r_a}{2} \frac{\pi}{T} \\ \Rightarrow \omega &= \sqrt{r_p r_a} \frac{r_p + r_a}{2r^2} \frac{2\pi}{T}. \end{aligned}$$

## 11.2. 위성 데이터

모행성	위성 이름	$GM$	자전주기 <sup>2</sup>	공전주기 <sup>3</sup>	원지점	근지점	Sphere of influence	위성 반지름
Kerbin	Mun Minmus							

## 11.3. 엔진 데이터

200 FuelUnit = 1t. (oxygen and liquid)

10000 FuelUnit = 1t. (Xenon) density: expected around  $1t/m^3$  (liquid) ,  $1.05949191932 m^3 = 1t$ .

density:  $0.657420 m^3$ ,  $0.525t$  xenon. (xenon)  $0.0406 > 0.0371 m^3$ ,  $0.04t$  Globally, total fuel mass : tank mass = 8:1. (round bare tank. extra structure may weigh more.) Globally(xenon), total fuel mass : tank mass = 14:11. (round bare tank. extra structure may weigh more.)

	Liquid/Oxydizer	Solid	RCS	Xenon	Ore
Density ( $t/m^3$ ) <sup>*</sup>	~1	~1	0.5~4 <sup>!</sup>	~1	1.3~ 2
Density (t/FuelUnit)	1/200	3/400	1/250	1/10000	1/100
Least Mass Ratio of (Tank Structure) (Fuel Capacity)	1/8	0.231~0.2439 <sup>†</sup>	>0.1333~0.1775	11/14	1/6
Cost per FuelUnit	0.8(Liq)/0.18(Oxy)	0.6	1.2	4	-

<sup>!</sup> Smaller tank can store more densely.

<sup>\*</sup> Measured from the graphic

<sup>†</sup> Including engine

## CHAPTER 12

### 프로토콜

#### 12.1. Kerbin 궤도 프로토콜



## 탄도학 기초 for Kerbal Propulsion Lab

De revolutionibus orbium coelestium et momentis vehiculorum liberorum

---

저자 Albertus Liberius Humaniensis Disciplinus

[twpf.jp/alblib](http://twpf.jp/alblib)

초판 초쇄 2016년 8월 20일

---

이 저작물은 크리에이티브 커먼즈 저작자표시-동일조건변경허락  
4.0 국제 라이선스에 따라 이용할 수 있습니다.



라이선스 본문: <https://creativecommons.org/licenses/by-sa/4.0/deed.ko>

본 라이선스에 따라 공유 및 기여를 할 수 있는 원문의 주소는 다음과 같습니다.

<https://github.com/alblib/keral-book>

단, 기초자료로서 활용된 Kerbal Space Program의 데이터는 Squad사의 저작물로서  
저작권 보호를 받습니다. (<https://kerbalspaceprogram.com/>)