**우주궤도역학 TermProject 2 Report**

**항공우주공학과**

**19011257 이동주**

**순서**

1. **알고리즘**
2. **코드 설명**
3. **알고리즘 설명**

**먼저 궤도를 그리기에 앞서, 필요한 값들을 생각해 보았습니다.**

1. **GroudTrack -> ECEF(x,y,z)를 이용하여 위도, 경도, 고도 계산**
2. **SkyView -> ENU frame의 좌표를 이용하여 Azimuth Angle, Elevation angle 계산**

**우선, 제가 알고 있는 값들은 t0, Semi-major-axis (a), Eccentriciy(e), Inclination angle (i), Argument of Perigee(omega), Mean Anomaly at t0 (M0), RAAN이 있었습니다. 그렇기에, Perifocal frame에서 위성의 좌표를 정의하기 위해선 True anomaly 값이 필요했기에, M0를 이용하여 true anomaly를 구했습니다.**

1. **원하는 시각의 t를 설정한 뒤에, M – M0 = n(t – t0)를 이용하여, M을 구한다. 여기서 n은 mean motion.**
2. **M = E – eSinE 를 이용하여 E, Eccentricity Anomaly 계산**
3. **True anomaly는 E에 관한 식으로 유도할 수 있기에, 계산**
4. **Perifocal frame에서 좌표를 계산**
5. **Ground Track의 경우, Perifocal -> ECI frame -> ECEF frame으로의 좌표변환을 통해 Geodetic frame에서의 좌표를 계산한 뒤 Plot.**
6. **SkyView 의 경우, Perifocal -> ECI -> ECEF -> ENU frame으로의 좌표변환을 통해, Elevation angle, Azimuth angle을 구한 뒤 plot.**

**이러한 알고리즘을 통해 Matlab상에서 코드를 완성했습니다. 다음 페이지에서 코드에 대한 자세한 설명을 이어 가겠습니다.**

**텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명**

**우선, nav.mat 파일을 matlab의 workspace로 가져온 뒤, 제가 원하는 날짜인 t를 datetime 형식으로 저장했습니다. 그리고, 24시간 동안 위성의 궤도를 추적하기 위해, 설정한 시간에서 24시간을 더한 t\_end를 생성했습니다. 또한, 위성 최소 양각의 El\_mask 와, 관측지점의 경도와 위도를 저장하는 코드를 작성했습니다.**

**텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명**

**그리고, 계산의 편의성을 위해 Semi–latus-rectum, P를 미리 계산 한 뒤, Mean Anomaly를 계산 하기 전 Mean Motion 도 각 위성별로 계산을 했습니다.**

**그리고, 반복문안에서, 시간이 경과함에 따라 값들의 변화를 차례대로 벡터에 저장하기 위해, 미리 Plot을 하기 위한 빈 벡터를 생성했습니다.**

**텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명**

**또한, 앞에서 말씀드린 M = n\*(t – t0) + M0를 계산하기 위해, fzero라는 내장함수를 사용하여 비선형식의 해를 수치해석적으로 계산하였습니다. 그렇게 각 위성별로 Eccentricity Anomaly를 계산하였고, 계산한 E를 이용하여 Trueanomaly를 atan2함수를 이용하여 계산했고, 또한 위성까지 거리의 크기 또한 계산했습니다.**

**텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명**

**그리고, Perifocal frame에서 ECI frame으로 변환하기 위한 회전행렬을 계산하였는데, 앞서 nav.mat 에서 얻어온 orbit parameter들, RAAN, Argument of Perigee, Inclination Angle을 이용하여 Rotation Matrix를 계산하였고, 그 회전행렬에 Perifocal frame상의 위치벡터를 곱해주어, ECI Frame에서의 위치벡터를 만들었습니다.**

**텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명**

**또한 ECI frame에서 ECEF Frame으로 변환해야 했기에, Siderealtime을 계산해야 했습니다. 그렇기에 기준시간을 2000년 1월 1일 정오로 설정한 뒤, 지구의 자전속도를 rad/sec 단위로 구하고, 현재시간에서 기준시간의 차이를 초단위로 계산하여, 기준시간에서의 LST에 지구자전속도 x 시간 만큼의 각도를 더 구해서 SideRealTime을 계산했습니다. 가시성을 높이기 위해 mod함수를 사용하여 0 ~ 360도 사이의 값으로 바꿨습니다. 그렇게 구한 Siderealtime을 이용해 ECEF로 변환하는 DCM을 만든 뒤, 위에서 구한 ECI frame에서의 위치벡터를 곱하여 ECEF에서의 위치벡터를 구하였고, 또한 matlab 내장함수인 ecef2geodetic 함수를 사용하여, 경도와 위도를 계산하였습니다. 이렇게 Ground Track을 그리기 위한 경도, 위도, 고도를 계산하여 plot할 준비를 마쳤습니다.**

**텍스트, 폰트, 스크린샷, 대수학이(가) 표시된 사진

자동 생성된 설명**

**그리고, SkyPlot을 위해 Elevation, Azimuth angle등이 필요했기에, ENU frame으로 변환하는 행렬을 만들었습니다. 또한, 바로 위성의 위치를 ENU frame으로 계산할 수 없기에, 관측지점인 Radar position을 ECEF상에서 정의한 R\_ecef를 계산하고, 상대적으로 위성의 위치를 ENU Frame에서 계산하여 RE,RN,RU를 벡터로 계산하였습니다.**

**텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명**

**이렇게 구한 ENU frame 상의 위성의 위치벡터를 이용하여, Azimuth angle과 Elevation을 계산했습니다. 앞서 위성 최소 양각을 설정했기에, 그보다 낮은 Elevation은 무시하였고, 또한 acos함수가 0도~180도에서 각을 반환하기에,**

**0도~360도 사이의 각을 얻기 위한 반복문을 만들었습니다.**

**텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명**

**그렇게 구한 모든 데이터들, Ground Track을 그리기 위한 Longitude와 Latitude, SkyView를 그리기 위한 Elevation과 Azimuth angle을, 분 단위로 계산 한 뒤, 앞서 만들었던 빈 벡터에 차례대로 저장해 주었습니다.**

**이렇게 Ground Track, SkyView를 만드는 코드를 Matlab을 이용하여 만들었습니다. 부족하지만 최선을 다해서 만들었습니다. 감사합니다!**