

10/10  
~ 10/12

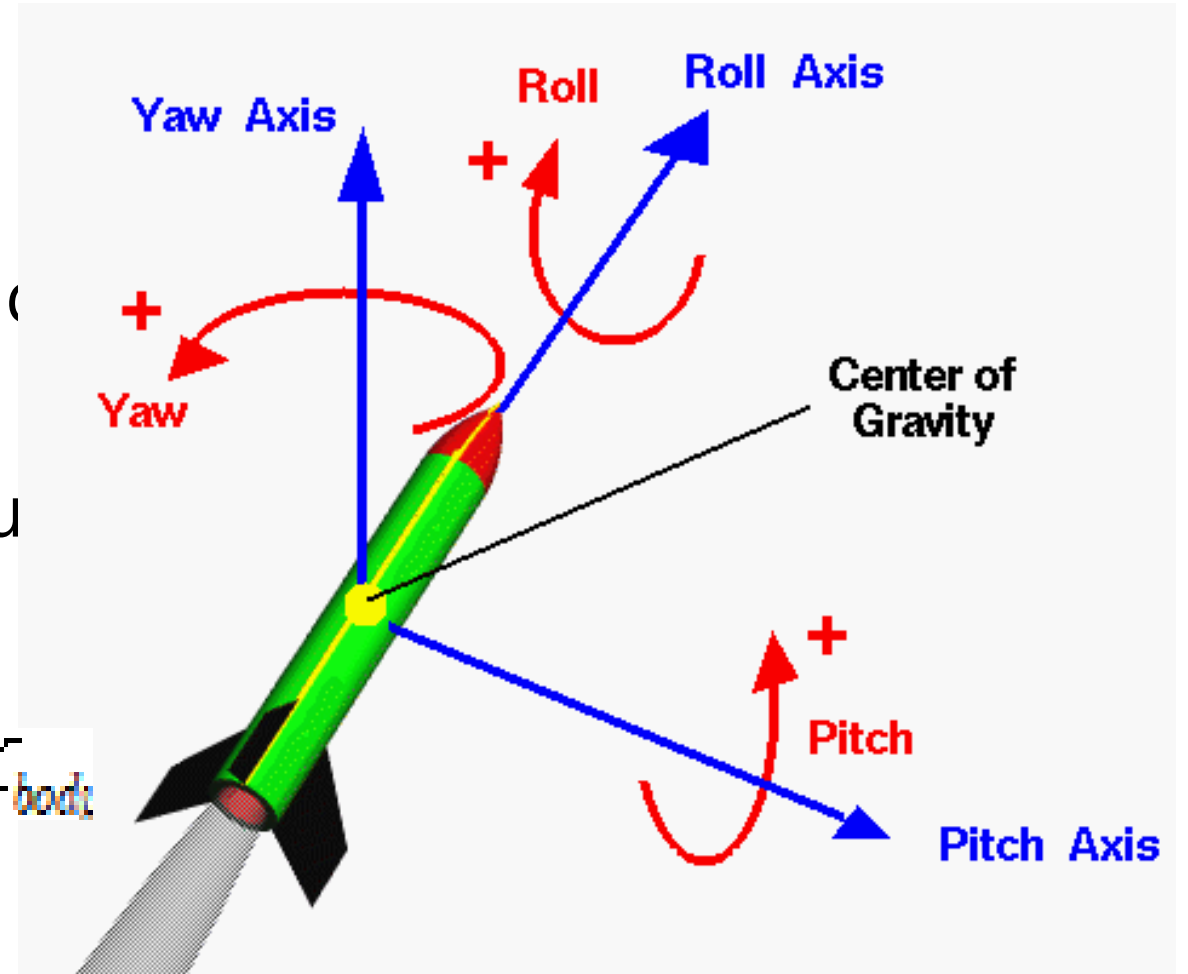
서울대학교 2017 가을축제

국도루팡

# 궤적의 기초 - 변수들(6DOF)

- 로켓은 공중에 붕붕 떠있음.
- $X, Y, Z$ (Position) +  $\phi, \theta, \psi$ (Attitude)
- $u, v, w$  (velocity) +  $p, q, r$  (angular velocity)

$$X_{ground} = R(\psi) \times R(\theta) \times R(\phi) \times X_{body}$$



# 로켓의 위치와 속도를 나타내기

- 지면 좌표(ECI)와 로켓 좌표(Body) 사이를 변환 하는 데에는 오일러 각, Quaternion 이 있음.
- 본 프로그램은 연산하는데 Quaternion, 결과를 보여주는데 Euler angle 사용.
  - Euler angle은 숫자만 봐도 대충 자세를 머릿속에 그릴 수 있을 때까지 익숙해질 것.
  - Quaternion은 왜 이것 쓰는지 정도만 이해하면 충분.

# Euler Angle 읽기

- Euler angle은 좌표계의 변환을 나타낼 때 쓴다.

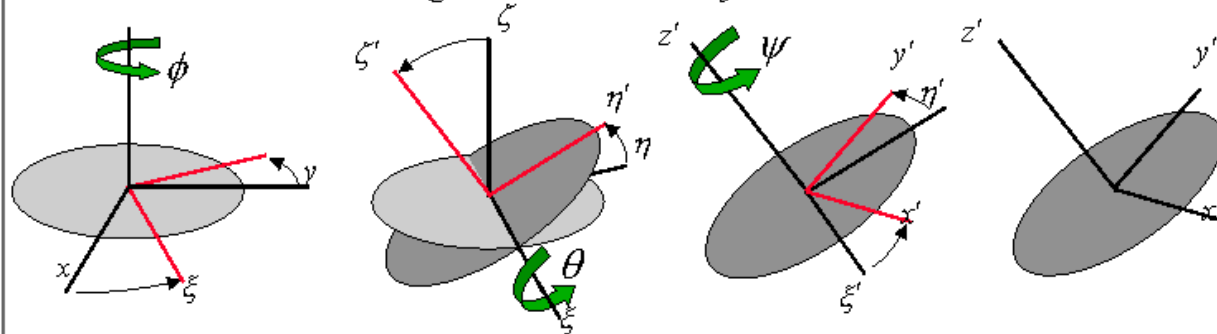
- X 축 기준으로 Phi (Roll)
- Y 축 기준으로 Theta (Pitch)
- Z 축 기준으로 Psi (Yaw)

- 우리 로켓은 Roll을 무시..

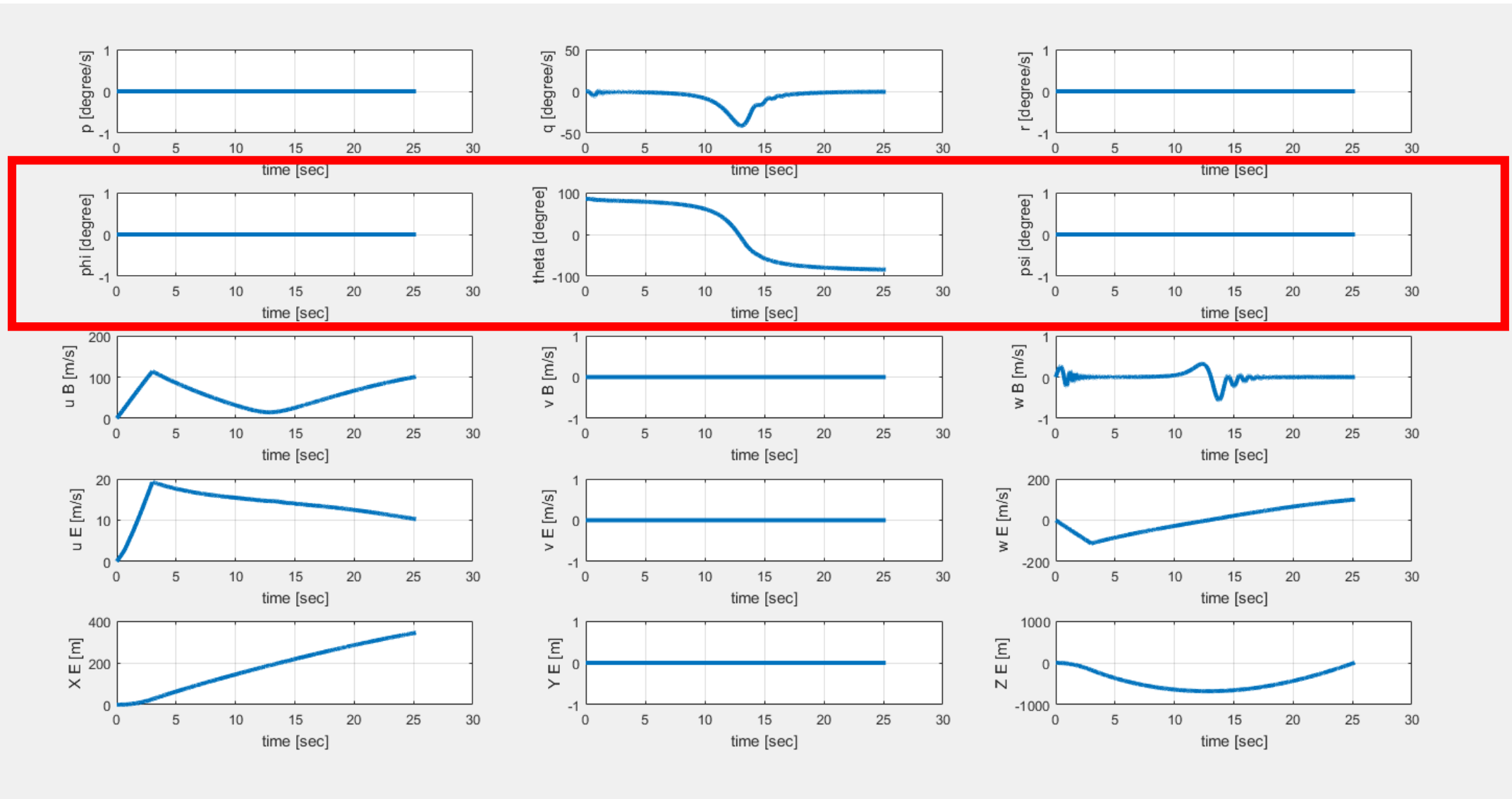
## Euler Angles

- The picture in 3D is similar:  $(x, y, z) \rightarrow (x', y', z')$
- Nine direction cosines
- Three independent coordinates specify orientation
- Euler angles are the conventional choice  $\Theta = \phi\theta\psi$

$z$  • three rotations give the simulation-frame orientation

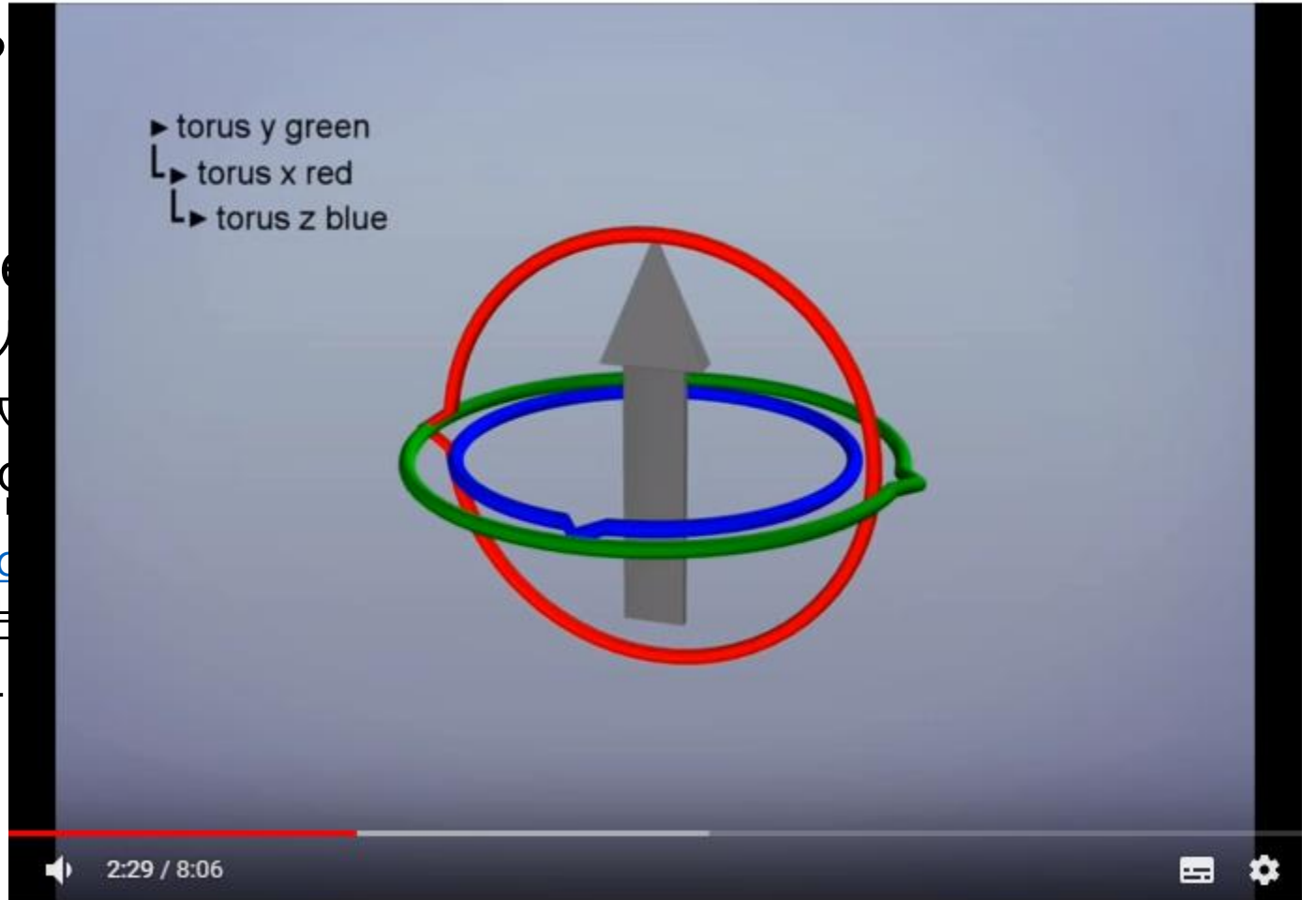


# 시뮬레이션의 자세 읽는법



# 왜 프로그램은 쿼터니안을 쓰는가?

- Euler angle은 편하고(?)
- 그럼에도 불구하고 Euler
  - 회전 순서를 언제나 명시
    - X-Y-Z? X-Z-Y? Z-X-Y? 자
  - Gimbal Lock 발생. 즉,  $\theta = 0$  일 때
  - <https://www.youtube.com/watch?v=Z0080000000>
  - 360도에서 불연속적으로 바뀌는
- -> Quaternion으로 계산하자



# Quaternion에 대한 짧은 소개

- 해밀턴이 고안해낸 4원수.
- 좌표계의 회전을 4개의 원소로 나타내자!

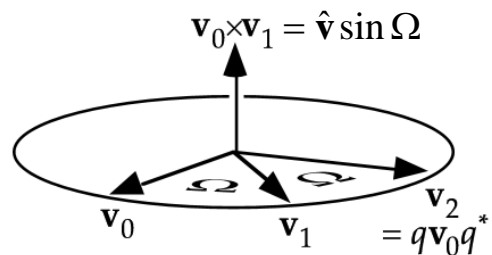
$$\mathbf{H} = \{(a, b, c, d) \mid a, b, c, d \in \mathbf{R}\}$$

- $q = a + b * \hat{i} + c * \hat{j} + d * \hat{k}$
- $H = \{w, x, y, z\} = \{w, v\}$

# Quaternion ★★

- ★**Theorem.** 3차원의 임의의 벡터 와 unit quaternion  $q = [\cos \Omega \quad \hat{\mathbf{v}} \sin \Omega]$  에 대해,  $\vec{\mathbf{x}}' = q\vec{\mathbf{x}}q^*$  는 벡터  $\vec{\mathbf{x}}$  를  $\hat{\mathbf{v}}$  을 축으로 하여  $2\Omega$  만큼 회전시킨다.

(proof.) 임의의 3차원 상의 벡터  $\vec{\mathbf{x}}$  는  $\hat{\mathbf{v}}$  와 평행인 성분과 그에 수직인 성분으로 표현할 수 있다. 따라서  $\vec{\mathbf{x}} = s\hat{\mathbf{v}} + r\hat{\mathbf{v}}_o$  ( $\hat{\mathbf{v}} \perp \hat{\mathbf{v}}_o$ ) 으로 표현할 수 있다. 따라서,  $\vec{\mathbf{x}}' = q\vec{\mathbf{x}}q^* = q(s\hat{\mathbf{v}} + r\hat{\mathbf{v}}_o)q^* = s(q\hat{\mathbf{v}}q^*) + r(q\hat{\mathbf{v}}_oq^*)$  가 된다.



①  $q\hat{\mathbf{v}}q^*$

$$\begin{aligned} q\hat{\mathbf{v}}q^* &= [\cos \Omega \quad \hat{\mathbf{v}} \sin \Omega][0 \quad \hat{\mathbf{v}}]q^* = [-\sin \Omega \quad \hat{\mathbf{v}} \cos \Omega]q^* = [-\sin \Omega \quad \hat{\mathbf{v}} \cos \Omega][\cos \Omega \quad -\hat{\mathbf{v}} \sin \Omega] \\ &= [0 \quad \hat{\mathbf{v}}] = \hat{\mathbf{v}} \end{aligned}$$

②  $q\hat{\mathbf{v}}_oq^*$

위의 그림과 같이  $\hat{\mathbf{v}}_o$  로 부터  $\Omega, 2\Omega$  만큼 돌아간 벡터를  $\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2$  라 하자. 그러면  $\hat{\mathbf{v}}_1\hat{\mathbf{v}}_0^* = [\hat{\mathbf{v}}_1 \hat{\mathbf{v}}_0^* \quad \hat{\mathbf{v}}_0 \times \hat{\mathbf{v}}_1] = [\cos \Omega \quad \hat{\mathbf{v}} \sin \Omega] = q$ ,  $\hat{\mathbf{v}}_2\hat{\mathbf{v}}_1^* = [\hat{\mathbf{v}}_2 \hat{\mathbf{v}}_1^* \quad \hat{\mathbf{v}}_1 \times \hat{\mathbf{v}}_2] = [\cos \Omega \quad \hat{\mathbf{v}} \sin \Omega] = q$  가 된다. 따라서,

$$q\hat{\mathbf{v}}_oq^* = (\hat{\mathbf{v}}_1\hat{\mathbf{v}}_0^*)\hat{\mathbf{v}}_o(\hat{\mathbf{v}}_2\hat{\mathbf{v}}_1^*)^* = \hat{\mathbf{v}}_1(\hat{\mathbf{v}}_0^*\hat{\mathbf{v}}_o)(\hat{\mathbf{v}}_1\hat{\mathbf{v}}_2^*) = (\hat{\mathbf{v}}_1\hat{\mathbf{v}}_1)\hat{\mathbf{v}}_2^* = -\hat{\mathbf{v}}_2^* = \hat{\mathbf{v}}_2$$

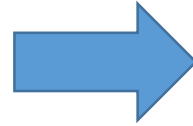
$$\implies \therefore \vec{\mathbf{x}}' = q\vec{\mathbf{x}}q^* = s\hat{\mathbf{v}} + r\hat{\mathbf{v}}_2$$



# 쿼터니언(계속)

- 이걸 쓰면, 자세의 변화를 \*쉽게\* 컴퓨터로 연산 가능.

$$\begin{aligned}\dot{\phi} &= p + q \sin \phi \tan \theta + r \cos \phi \tan \theta \\ \dot{\theta} &= q \cos \phi - r \sin \phi \\ \dot{\psi} &= q \sin \phi \sec \theta + r \cos \phi \sec \theta\end{aligned}$$



-쿼터니언 방정식

$$\dot{e}_0 = -\frac{1}{2}(e_1 p + e_2 q + e_3 r) + k \lambda e_0$$

$$\dot{e}_1 = \frac{1}{2}(e_0 p - e_3 q + e_2 r) + k \lambda e_1$$

$$\dot{e}_2 = \frac{1}{2}(e_3 p + e_0 q - e_1 r) + k \lambda e_2$$

$$\dot{e}_3 = \frac{1}{2}(-e_2 p + e_1 q + e_0 r) + k \lambda e_3$$

$$\lambda = 1 - (e_0^2 + e_1^2 + e_2^2 + e_3^2),$$

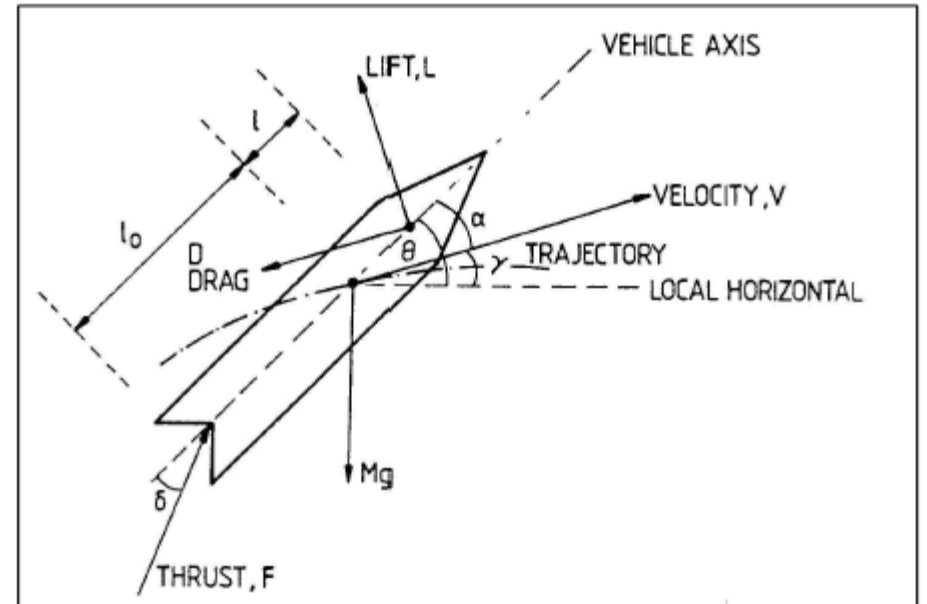
$k = \text{small enough}$

- $e_0 \rightarrow e_0 + \frac{de}{dt} dt$
- de를  $\langle p, q, r \rangle$ 과 초기  $e_0$ 로 계산
- 우리가 볼땐 Euler angle, 계산은 Quaternion으로 진행.

# 궤적을 만들어 봅시다

- Rocket Equation

$$\begin{aligned} V(t) &= \Delta V_{ideal} - \Delta V_g - \Delta V_D \\ &= g_0 I_{sp} \ln \left[ \frac{M_0}{M_f} \right] - \int_0^t g \sin \gamma dt - \int_0^t \frac{D}{M} dt \end{aligned}$$



# 궤적에 영향을 주는

- V ideal

- $M_0 / M_f$  : 보통 역수인  $M_f/M_0$ 를 효과적이다.

- 우주왕복선 SRB의 경우  $MR = 0.0$
- SNUKA-2는  $MR$ 이 0.85(...)
- SIM에는  $M_0$ ,  $M_f$ ,  $M_{pay}$ 를 정해주

- $I_{sp}$  : 범이형이 얘기해줬죠? 클수록
- $I_{sp}$ 를 쓰거나 Thrust (N)를 쓰거나

```
71 - Thrust=250;      % Thrust [N]
72 - tb=3;          % Burn time (sec)
73 - time_f=25.2;   % Final time for trajectory calculation (sec)
74
75 %Mass
76 - m0=5.23;       % Initial mass (kg)
77 - mf=4.86;       % Final mass (kg)
78 - mp=m0-mf;      %propellant mass(kg)
79
80 - Launch_angle=85; % Launch angle [Degree]
81
82
83 - mpl=mf;         %payload mass(kg) %무시
84 %%MR 사용시
85 %{
86   MR=0.34;
87   lsp=230;
88   g0=9.81;
89   m_dot=Thrust/lsp/g0;
90   m0=m_dot*tb/(1-MR);
91   mf=m0-Thrust/(lsp*g0)*tb;
92 %}
93
94 %Moment of Inertia
95 %ni=1000.5;
96 - ni=1.0;         % 최대한 크게 하여 자세변화에 따른 궤적 영향력 줄임
97
98 - lx=.0155*ni;    %kg m^2
99 - ly=.64*ni;
100 - lz=.64*ni;
```

# 궤적에 영향을 주는 변수

- $V_g$ 
  - 중력손실. 단적으로,  $\gamma = 0$  (지면과 수평비행)이면 중력손실은 0.
  - 궤도에 올리는 발사체의 경우 중력손실이 중요.
  - IREC은 거의 수직으로 쏘아올린다 생각하면 됨.
- => 궤적프로그램에선 고려하지만, 우리가 고려할 필요는 없음.

# 궤적에 영향을 주는 변수

- V Drag

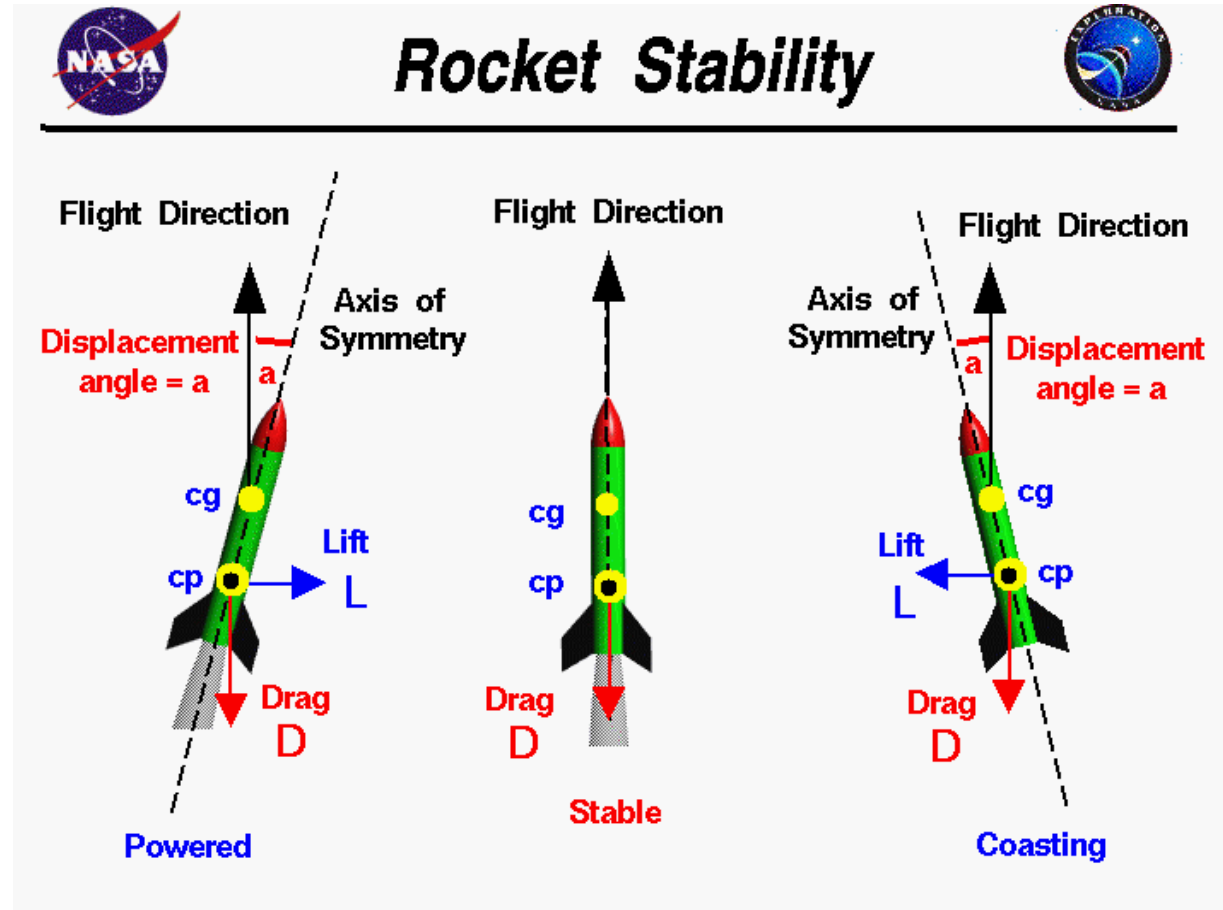
- 단순 저항뿐만 아니라
- 넓이, 공력계수를 통해

- $C_l$  – 진행방향과 동체 사
- $C_d$  – 진행방향 반대방향
- $C_{dy}$  – 진행방향 '옆면'의

```
121  
122 - S=pi/4*0.114^2; %Area (m^2)           %로켓직경고려  
123 - Sy=0.114*0.86;   %Area (m^2)         %로켓옆면적고려  
124  
125 %공력데이터의 정확한 값은 풍동실험으로 구함, 또는 대략 Missile Dotcom 사용  
126 - cd=0.35; %0.15 ~ 0.4  
127 - cdy=0; %0.01;  
128 - cl=0.001;  
129 - cla=1/pi*180; % PID 중 P 에 해당  
130 - cdyb=-1/pi*180; %(-)  
131 - rho=1.23; % Initial density (kg/m^2)  
132
```

# Static Margin (SM)

- 모든 공력은 공력중심에 작용
- 추력과 자세는 무게중심으로 작용
- 정적 안정성을 나타내는 기준: Static Margin.
- $(C_p - C_g)/\text{Diameter}$
- SNUKA-2: 2.0 Caliber.



# Simulation Process

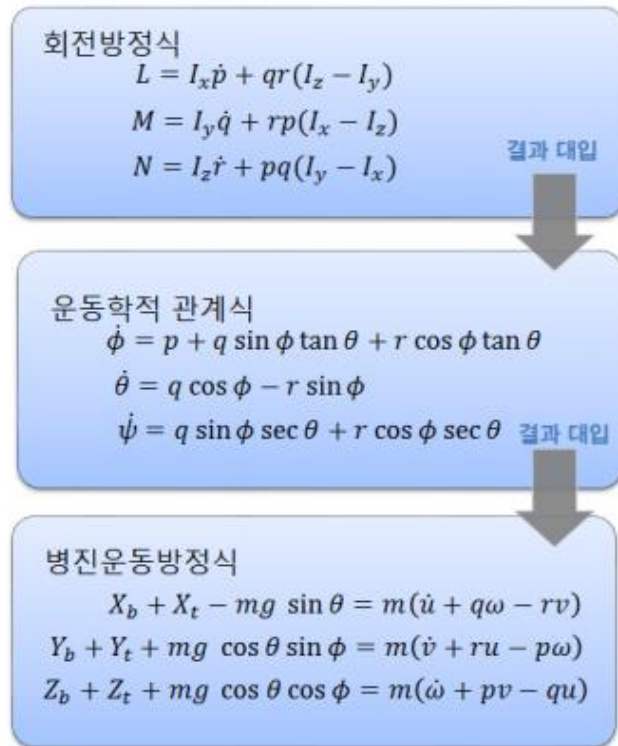
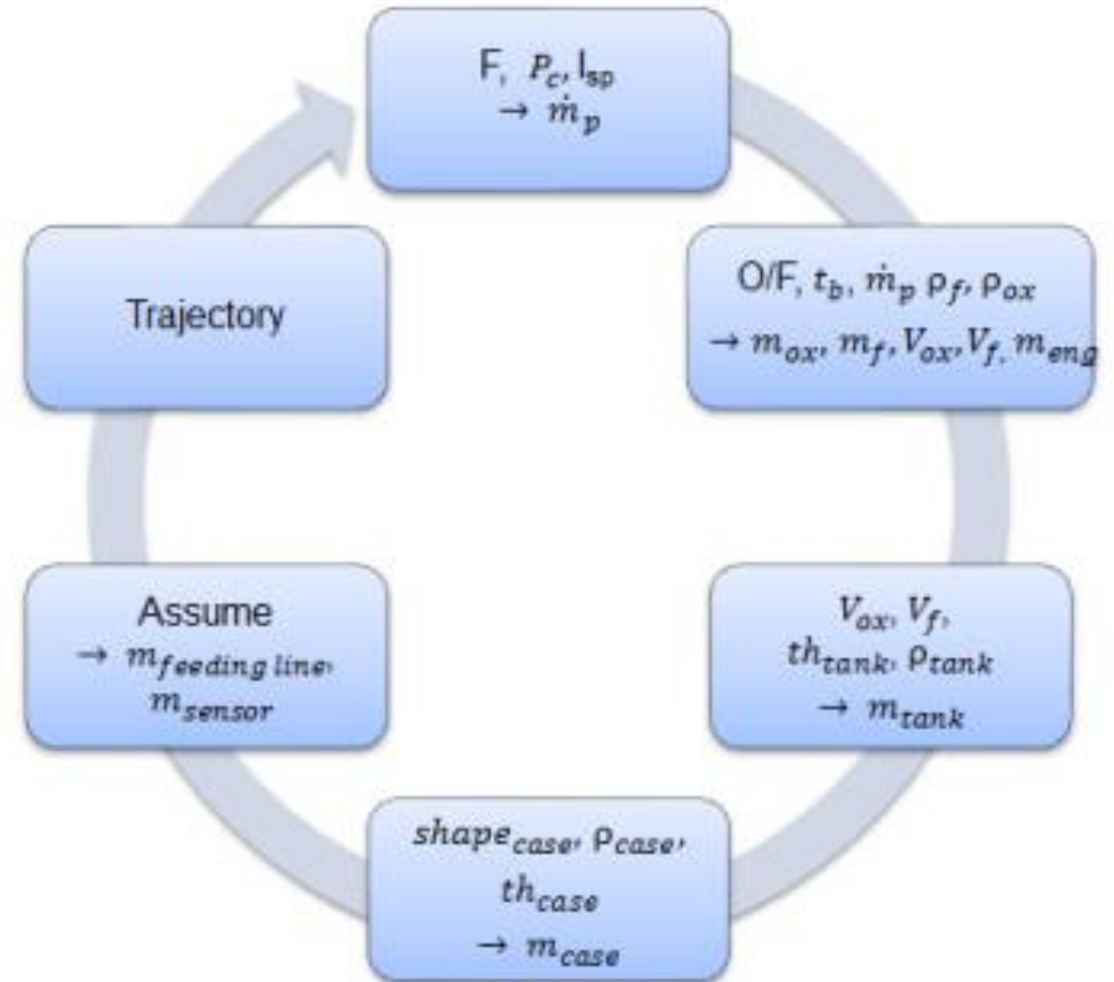


Fig. 7 Trajectory calculation algorithm



# 그래서 전 뭘 하면 됩니까?

- Goal: 고도 3km를 달성하는 고체모터와 mass cost를 대략 정하기.
- COTS 모터 사이트의 여러 모터들의 추력 profile을 보고, IREC에서 사용(고도가 2.5 ~ 3.5km)가능한 놈을 찾아라.
- 동시에 MR 또는 전체적인 질량, 면적, caliber 등의 허용범위를 지정해주길 바람.



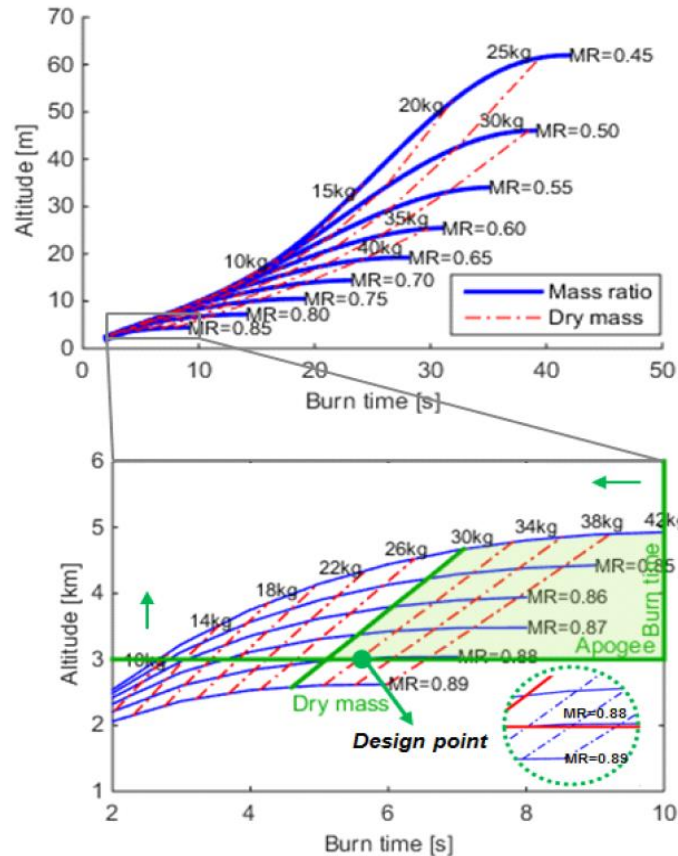


# Initial Design Condition

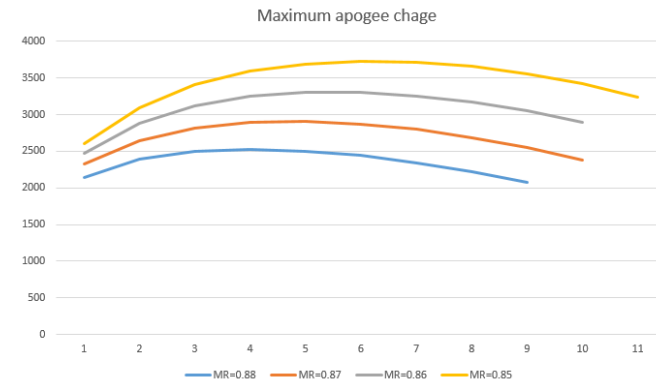
# Rocket design initial parameter

Simulation Condition	
Target apogee	10,000ft (3km)
Thruster type	Hybrid (H2O2/PE)
Number of stage	1
Engine clustering	no
Payload	4 kg
Recovery	Parachute

# Thrust deciding - Simulation result



Maximum altitude of the **2000 N** sounding rocket



Maximum altitude of the **2500 N** sounding rocket

Parameter	Value
MR	0.8645
Burning time	6 s
Mass	46.5 kg
Max apogee	3007.6 m

# Rocket design condition

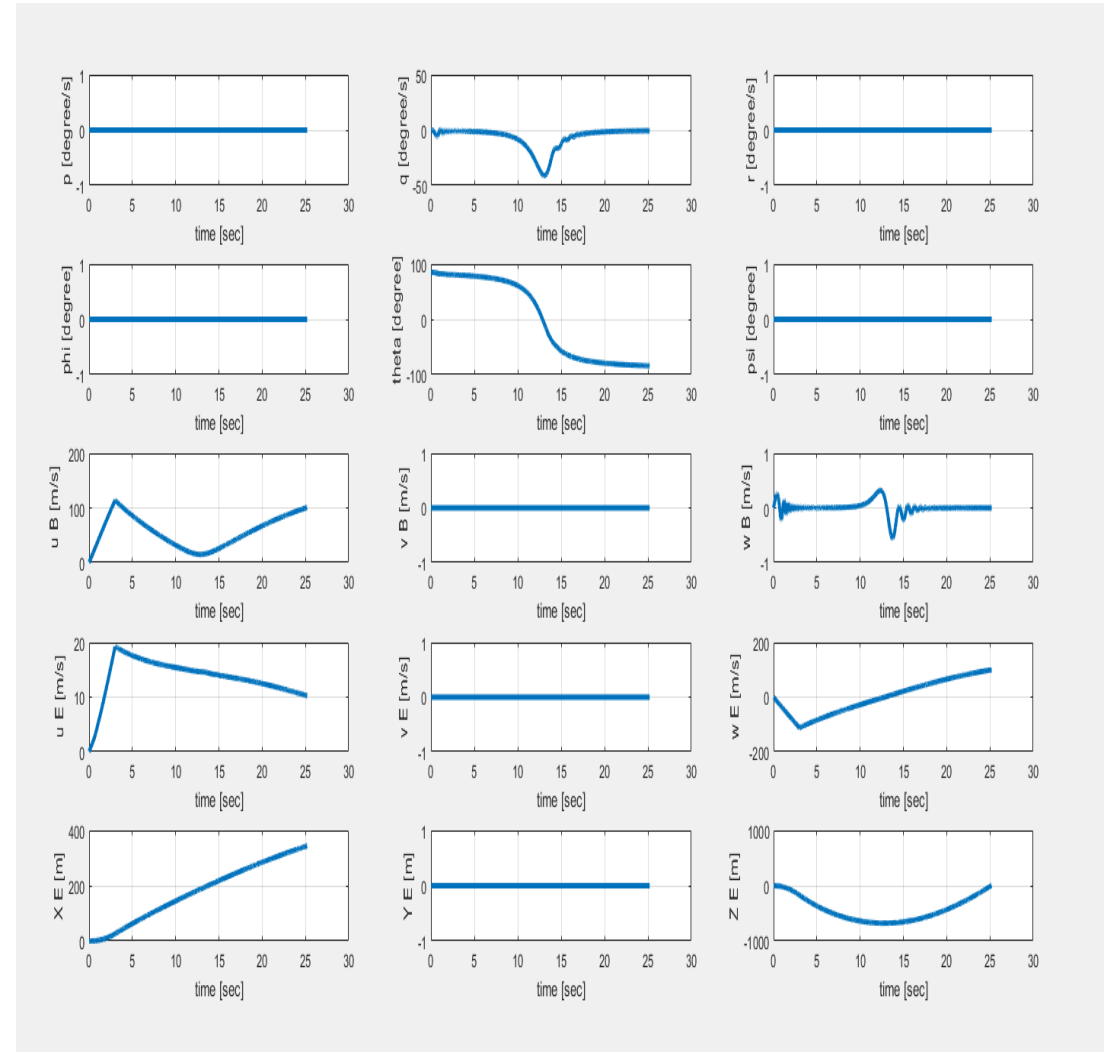
Stage	Mass
Nose Cone	1 kg
Avionics	2 kg
Recovery system	5 kg
Tank	4.5 kg
Valve	4 kg
Propulsion system	14 kg
Propellant	6.5 kg
Fins	1 kg
Rocket body	8 kg
Total	46.0 kg

Simulation Condition	
Thrust	2500 N
Mass ratio	0.8645
Burn time	6 seconds
Chamber Pressure	20 bar
Launch angle	85 deg
Diameter	200 mm
L/D	12.5
Acceleration	5.2 G

# Simulation 결과를 해석하자

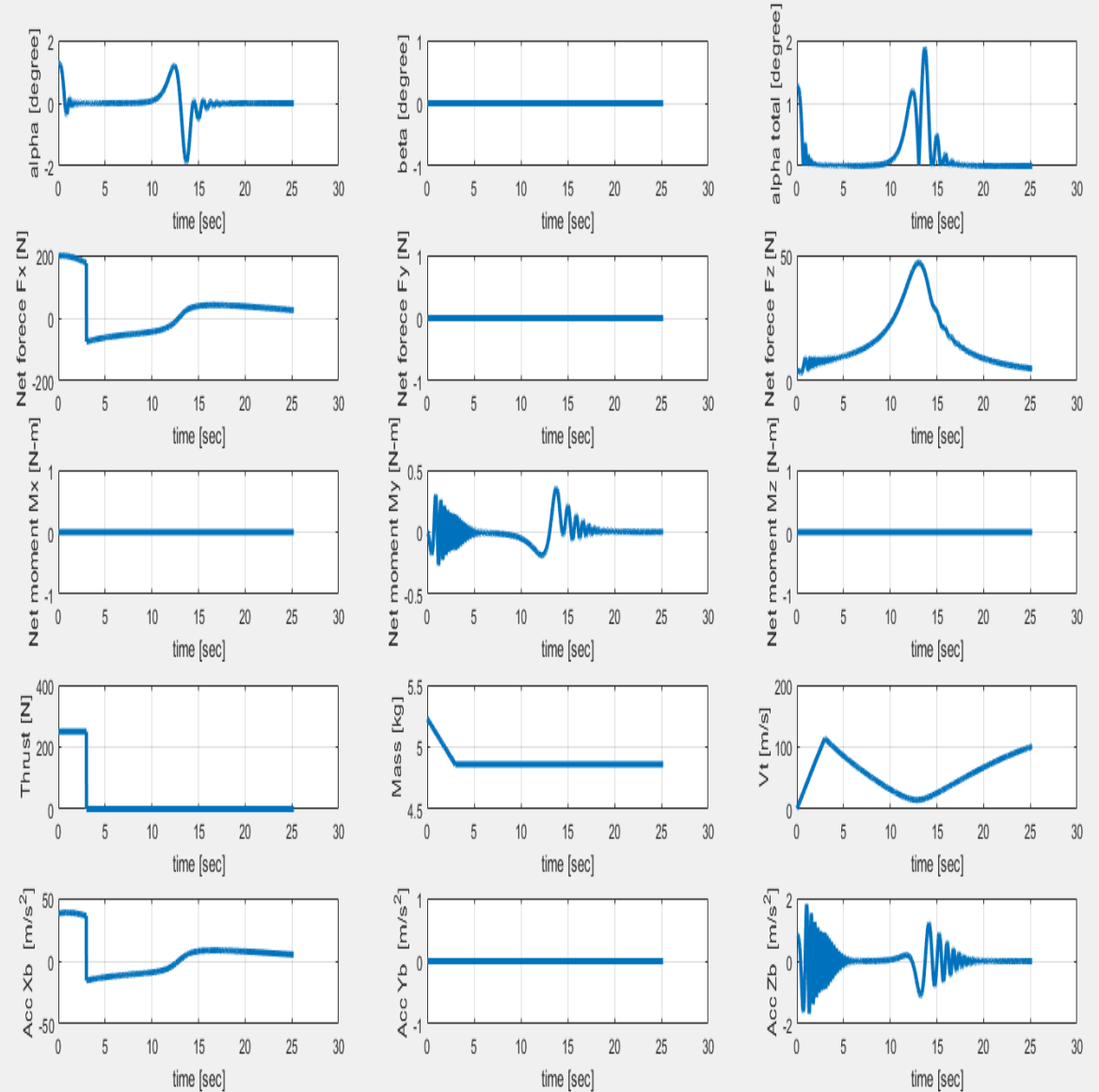
※ Kaist-Sim 원본 코드결과임

- Figure(1)
- 비행 결과 '측정치'들을 변수로 출력
- 위에서부터:
- 각속도
- 자세
- 속도(body)
- 속도(Earth)
- 위치(Earth) – z축은 상하반전

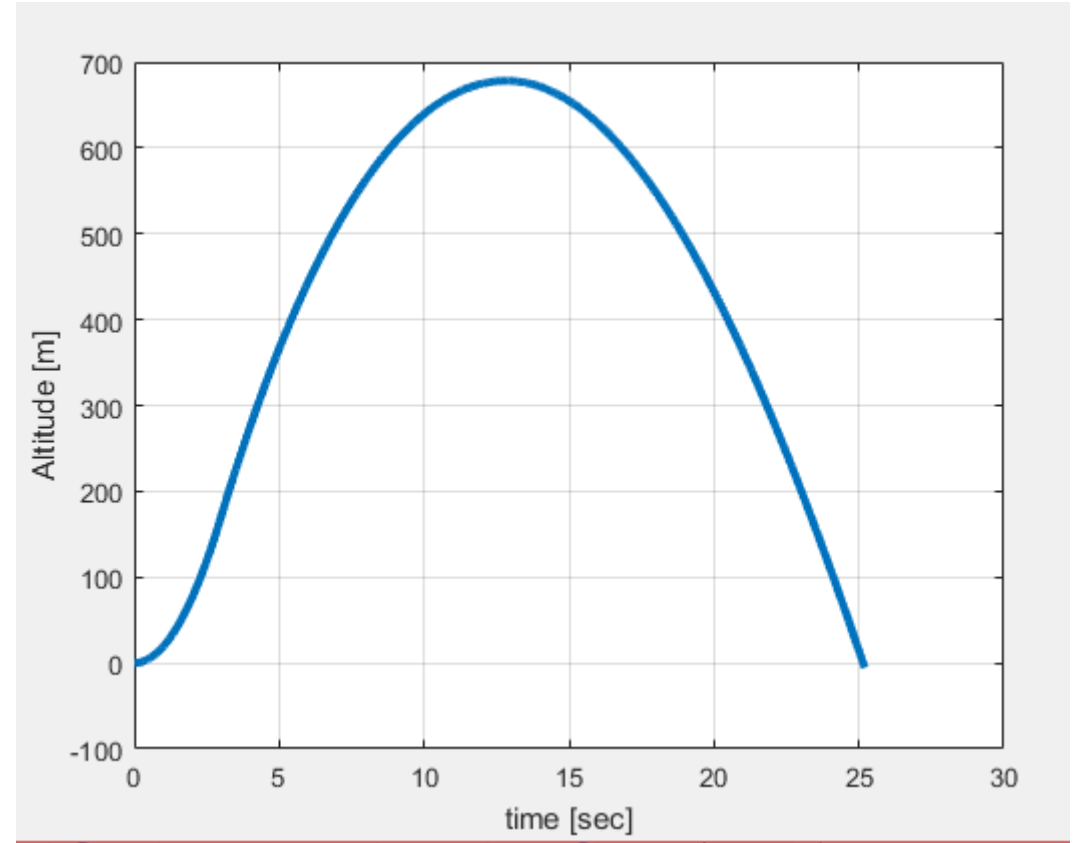
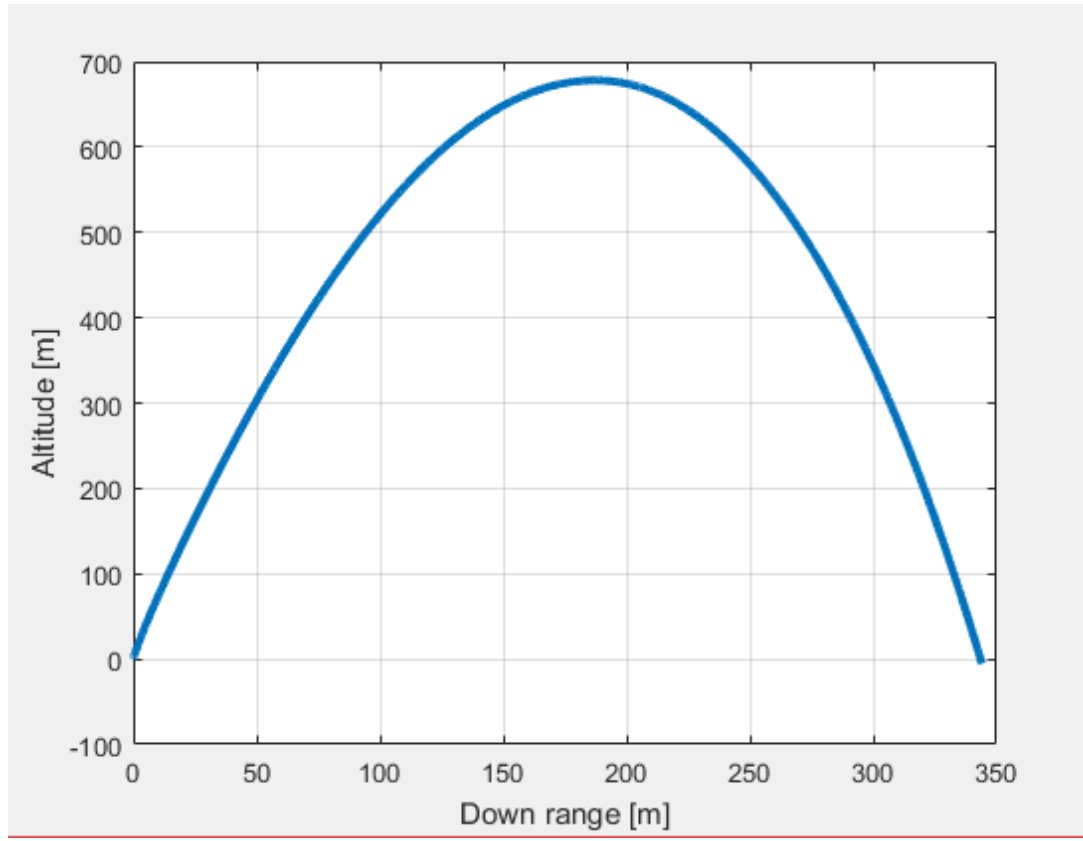


# Simulation 결과를 해

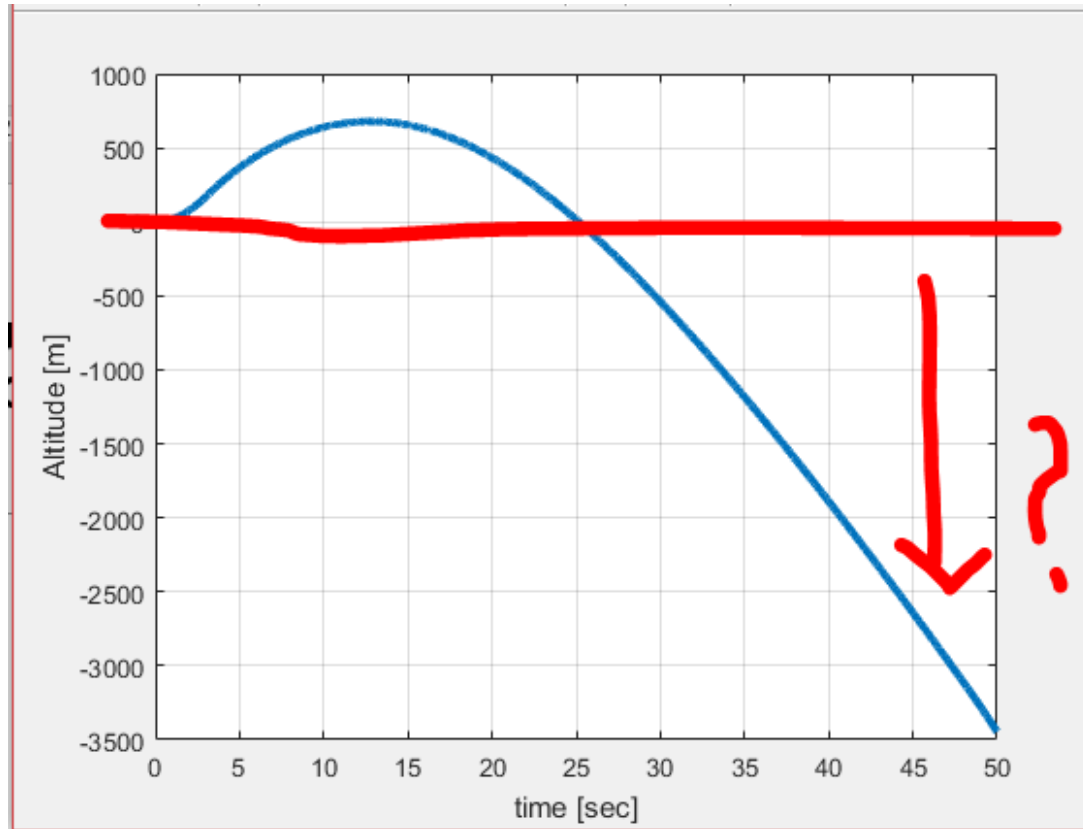
- Figure(2)
- 비행 결과 역학데이터를 표시
- 받음각/편각/전체 각도
- Body Frame 힘
- Body Frame 모멘텀
- 추력/질량/속력
- Body Frame 가속도



# Simulation 결과를 해석하자



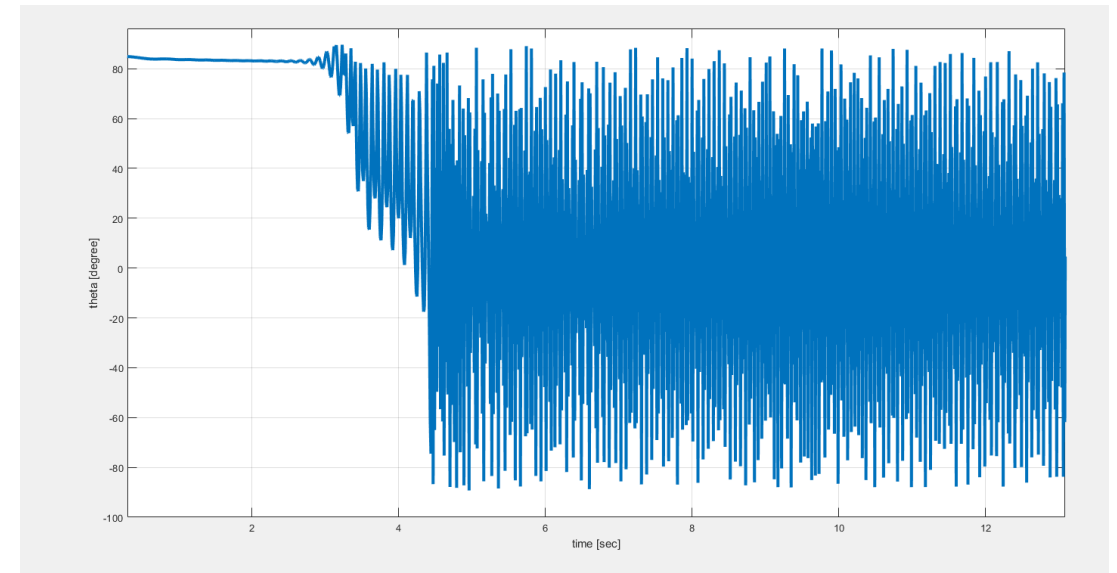
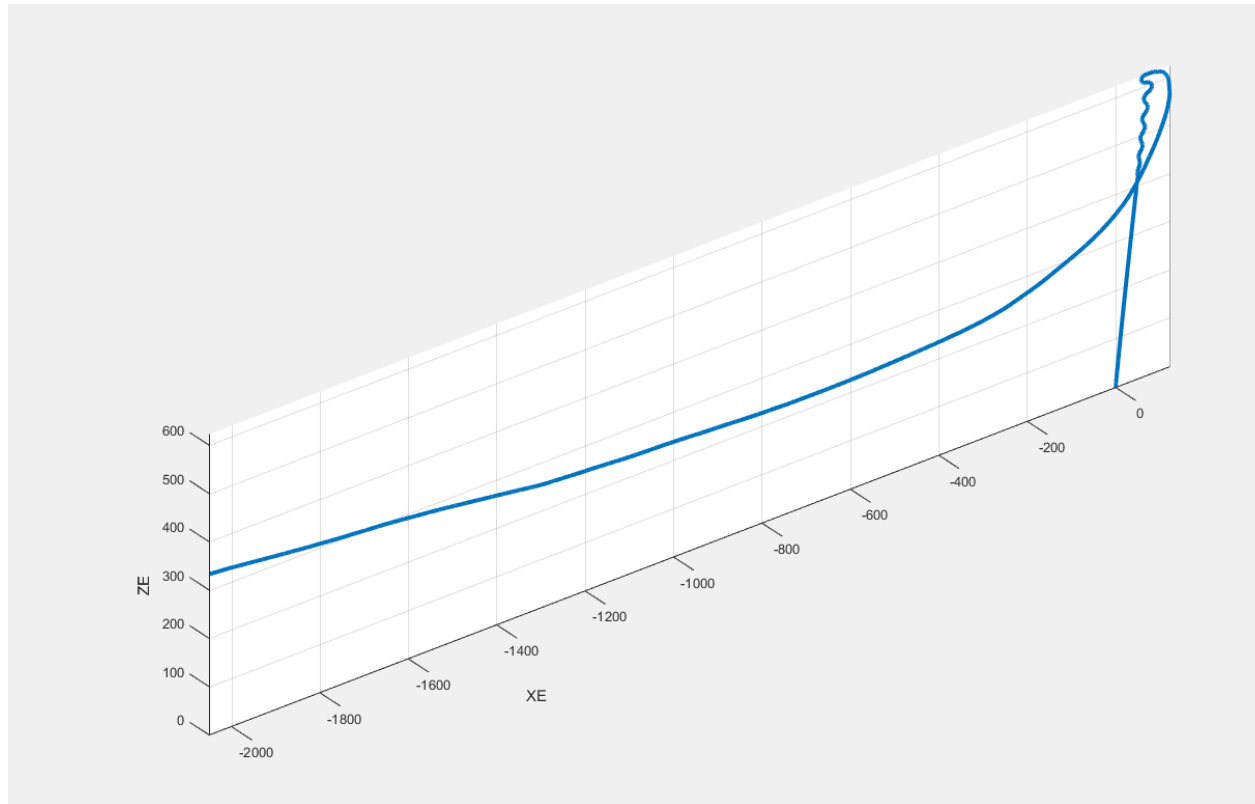
# 이상한 결과가 나온 경우



```
68  
69 %*****INPUT*****  
70  
71 Thrust=250; % Thrust [N]  
72 tb=3; % Burn time (sec)  
73 time_f=50; % Final time for trajectory calculation (sec)  
74
```



# 이상한 결과가 나온 경우



# 참고자료(달라하면 언제든 줌)

- 과산화수소/폴리에틸렌 하이브리드 로켓엔진을 이용한 과학로켓 개념설계와 비행궤적 계산
- 과학로켓 설계 기초방향
- 소형 사운드 로켓 적용을 위한 H<sub>2</sub>O<sub>2</sub>/PE 하이브리드 로켓 시스템