



J. Korean Soc. Aeronaut. Space Sci. 50(10), 709–716(2022)

DOI: <https://doi.org/10.5139/JKSAS.2022.50.10.709>

ISSN 1225-1348(print), 2287-6871(online)

## 위성발사체 상단의 고체로켓모터 유도를 위한 Gamma 유도 알고리즘 성능 분석

송은정<sup>1</sup>, 조상범<sup>2</sup>, 선병찬<sup>3</sup>

### Performance Analysis of the Gamma Guidance Algorithm for Solid Rocket Kick Motors of Upper Stages of Space Launch Vehicles

Eun-Jung Song<sup>1</sup>, Sangbum Cho<sup>2</sup> and Byung-Chan Sun<sup>3</sup>

Korea Aerospace Research Institute, Daejeon, Republic of Korea

#### ABSTRACT

In this paper the Gamma guidance law, which was used for IUS (Inertial Upper Stage), is applied for solid-motor guidance of a upper stage of a satellite launch vehicle. The RCS (Reaction Control System), which activates after burnout of the upper stage, is employed for the convergence of the guidance algorithm and compensation of velocity errors induced by the solid motor. The algorithm is also simplified by replacing the time-consuming numerical integration process to predict final vehicle states with Keplerian trajectories. The performance of the algorithm is evaluated by conducting 3-DOF computer simulations for off-nominal flight conditions. The numerical results show that Gamma guidance can reduce the orbit injection accuracy in comparison with that obtained by applying open-loop commands.

#### 초 록

본 논문에서는 IUS (Inertial Upper Stage)에 사용된 Gamma 유도 방식을 상단이 고체모터로 구성된 위성발사체 유도를 위해 적용해 보았다. 알고리즘의 수렴성 및 고체모터의 속도 오차 보상을 위해 RCS (Reaction Control System)가 최상단 연소 종료 후 작동하는 것으로 두었다. 알고리즘 계산 과정 중 최종 궤도 투입 오차 예측을 위한 적분과정을 Keplerian 궤적으로 단순화함으로써 수치 적분과정이 없도록 알고리즘을 구성하였다. 유도 알고리즘 평가를 위해 비공칭 비행 조건에 대한 3-DOF 컴퓨터 시뮬레이션을 수행해 주어진 상단 유도 방식은 개루프 명령을 적용한 경우 대비 궤도 투입 오차를 줄일 수 있음을 보였다.

**Key Words** : Space Launch Vehicle(위성발사체), Guidance Algorithm(유도 알고리즘), Closed-Loop Guidance(폐루프 유도), Kick Motor(킵모터), Gamma Guidance(Gamma 유도), 3-DOF Simulation(3-자유도 시뮬레이션)

#### 1. 서 론

본 논문에서는 위성발사체의 최종단이 고체로켓모터로 구성되어 있는 경우 정확한 목표 궤도 투입 위한 Gamma 유도 알고리즘에 대한 분석을 수행하였다. 고체 엔진의 경우 액체 엔진과는 달리 전체 속도 증분을 비행 중 제어할 수 없기 때문에 궤도 투입 오차가 크게 발생할 수

있다. 미국의 보잉사에서 개발한 IUS의 경우 임무에 따라 1~3개까지 다양한 고체모터로 구성이 가능한데, 고체모터의 앞서 언급한 단점을 보완하기 위해 저추력의 액체 버니어 엔진을 사용하여 고체모터에 의해 발생하는 속도 오차를 보상함으로써 궤도 투입 정밀도를 향상시켰다[1-3]. IUS는 1982~2004년까지 지구저궤도에서 고궤도까지 위성을 투입하는 다양한 임무를 수행하였다. 미

† Received : June 9, 2022    Revised : July 22, 2022    Accepted : August 10, 2022

<sup>1-3</sup> Principal Researcher

<sup>1</sup> Corresponding author, E-mail : [ejsong@kari.re.kr](mailto:ejsong@kari.re.kr)

© 2022 The Korean Society for Aeronautical and Space Sciences

공군의 Titan T34D의 상단으로 사용되어 지구정지궤도에 위성을 투입한 바 있으며, Space Shuttle의 탑재 베이에 탑재되어 목성 탐사를 위한 Galileo 위성을 지구 탈출 궤도에 투입하는 데에도 성공적으로 사용되었다. IUS 외에도 4단형 고체 엔진을 사용하는 인도의 ASLV (Augmented Satellite Launch Vehicle)의 경우도 고체 엔진에 의한 속도 증분이 부족한 경우 Velocity Augmentation System을 작동하여 속도 손실을 보상했다 [4]. 일본의 가장 최신의 고체 엔진 발사체인 Epsilon 발사체는 3단형으로 구성되어 있는데 액체 추진 시스템 Post Boost Stage (PBS)가 임무에 따라 최종단에 추가로 장착 가능하도록 구성되어 있다. PBS는 속도 조절 기능이 있어 궤도 투입 정밀도와 다양한 임무에 대응할 수 있는 임무 유연성을 향상시키기 위해 사용된다 [5]. 그러나 이런 속도 보상 시스템은 발사체 중량을 증가시켜 페이로드 질량이 줄어든다는 단점이 있기 때문에 발사체의 임무에 따라 필요성에 대한 검토가 요구된다.

IUS의 다양한 임무 수행을 위해 Boeing사의 Hardtla는 Gamma 유도 알고리즘을 개발하였다 [1-3]. Gamma 유도방식은 상태공간 모델에 대한 현대제어이론과 선형 대수기법을 사용하여 원하는 목표 궤도를 달성하기 위한 추력방향과 연소시점을 비행 중 반복계산을 통해서 결정하게 된다. 참고문헌 [6]에서는 알고리즘이 수렴하는 않는 경우에 대해서 분석했으며, 주로 속도 오차를 보상할 수 없는 조건에 대해 알고리즘이 발산해 Pseudo RCS를 제안하기도 했다. Pseudo RCS는 실제 RCS는 없지만 알고리즘의 수렴성을 위해 상단 연소 종료 후에 RCS가 있는 것처럼 구성하여 유도 문제를 푸는 경우를 의미한다.

새로 개발되는 발사체의 경우 페루프 유도 방식의 복잡성 때문에 목표 궤도 투입 정밀도 요구 조건에 제약이 크지 않을 경우 상단에서도 개루프 유도 방식을 적용하는 경우도 있다. 본 논문에서는 가장 단순한 형태의 유도 명령인 개루프 명령과 비행 중 반복 계산을 수행해야 하는 Gamma 유도 방식을 주어진 위성 발사체의 최종단에 적용해 유도 성능을 비교해 보았다. 먼저 주어진 위성발사체에 적용되는 Gamma 유도 알고리즘에 대해서 자세히 기술함으로써 알고리즘의 원리와 복잡성을 파악할 수 있도록 했다. 알고리즘의 수렴성을 위해 최종단 연소 종료 후 속도 오차 보상 시스템이 있는 경우로 가정하였으며, 유도 알고리즘 계산 단계 중 최종상태벡터를 예측하는데 필요한 적분 과정을 Keplerian 궤적을 사용해서 상태벡터를 계산함으로써 알고리즘 중에 적분과정이 없도록 단순화하였다. 다음으로는 주어진 위성발사체의 최종단 이전까지 누적된 위치 및 속도 오차에 대한 3-DOF Monte Carlo 시뮬레이션을 수행해 유도 방식에 따른 궤도 투입 성능을 평가하였다.

## II. 유도 알고리즘

본 절에서는 유도 알고리즘을 얻는 데 필요한 로켓의

운동방정식, 선형화된 운동방정식을 기술하고 주어진 위성 발사체 상단 유도를 위해 적용되는 유도 알고리즘 계산 과정에 대해서 기술하였다.

### 2.1 운동방정식의 단순화

공기력 영향을 무시할 수 있는 우주발사체 상단 구간에서 관성좌표계에서의 로켓 점질량모델에 대한 운동방정식은 다음과 같이 주어진다.

$$\ddot{x} = \begin{bmatrix} \ddot{r} \\ \ddot{v} \end{bmatrix} = \vec{f}(x, \lambda) = \begin{bmatrix} \vec{v} \\ g + a(t)u_T \end{bmatrix} \quad (1)$$

여기서  $\vec{r}$ ,  $\vec{v}$ 는 각각 발사체의 위치와 속도,  $a(t)$ 는 추력에 의한 가속도를 나타내고,  $\vec{\lambda}$ 는 목표궤도 투입을 위한 제어변수 벡터를 나타내는데 추력벡터 방향  $\vec{u}$ 와 점화시간  $t_{ig}$ 가 여기에 해당된다.  $\vec{g}$ 는 중력 벡터로 IUS의 경우  $J_2$ 항까지 고려된 중력모델을 사용하였다 [1].

$$\vec{g} = -\mu \left\{ 1 + \frac{3}{2} J_2 (r_E/r)^2 [1 - 5(\vec{u}_z \cdot \vec{r})^2 / r^2] \right\} \vec{r} / r^3 - 3\mu J_2 (r_E/r)^2 (\vec{u}_z \cdot \vec{r}) \vec{u}_z / r^3 \quad (2)$$

여기서  $\mu$ 는 지구 중력 상수,  $r_E$ 는 지구 장반경이고  $\vec{u}_z = [0 \ 0 \ 1]^T$ .

Gamma 유도법칙에서는 이점 경계치 문제를 쉽게 풀 수 있도록 추력구간을 순간적으로 속도를 얻는 것으로 가정하는 속도 임펄스로 근사시켰다. 속도 임펄스가 적용되는 시점은 일정시간 동안의 KM(Kick Motor) 연소 후에 얻어지는 위치와 임펄스를 적용했을 때 얻어지는 위치가 같아지도록 다음과 같이 정해진다. 위 운동방정식으로부터 연소종료 시점  $t_{BO}$ 에서의 위치벡터는 다음과 같이 주어지고

$$\vec{r}_{BO} = \iint_{t_{ig}}^{t_{BO}} \{g + a(t)u_T\} dt^2 + \int_{t_{ig}}^{t_{BO}} v(t_{ig}) dt + \vec{r}(t_{ig}) \quad (3)$$

$t = t_1$ 에서 속도 임펄스  $\Delta v$ 로 KM에 의한 속도이득을 모사할 경우 위치벡터는

$$\vec{r}_{BO} = \iint_{t_{ig}}^{t_{BO}} g dt^2 + \int_{t_1}^{t_{BO}} \Delta v dt + \int_{t_{ig}}^{t_{BO}} v(t_{ig}) dt + \vec{r}(t_{ig}) \quad (4)$$

연소시간이 충분히 짧아서  $\vec{r}_{BO} \approx \vec{r}_{BO}$ 로 두면

$$\iint_{t_{ig}}^{t_{BO}} a(t)u_T dt^2 = \int_{t_1}^{t_{BO}} \Delta v dt = \Delta v (t_{BO} - t_1) \quad (5)$$

$\Delta v = \Delta v u_T$ 로부터

$$t_1 = t_{BO} - \frac{\iint_{t_{ig}}^{t_{BO}} a(t) dt^2}{\Delta v}, \quad \Delta v = \int_{t_{ig}}^{t_{BO}} a(t) dt \quad (6)$$

$$\Delta t_1 \equiv \frac{\iint_{t_{ig}}^{t_{BO}} a(t) dt^2}{\Delta v} \text{로 두면}$$

$$t_{ig} = t_1 - (\Delta T_B - \Delta t_1), \quad \Delta T_B = t_{BO} - t_{ig} \quad (7)$$

따라서 원래의 운동방정식은 다음과 같이 p개의 임펄스로 구성된 운동방정식으로 변형할 수 있다. 즉 무추력 구간-임펄스-무추력구간-임펄스로 궤적을 단순화시킬 수 있다.

$$\dot{x} = \begin{bmatrix} v + \Delta v_i \\ g \end{bmatrix}, \quad i = 1, 2, \dots, p \quad (8)$$

### 2.2 운동방정식의 선형화

유도 문제는 원하는 목표궤도 조건을 만족시키는 비행 궤적을 생성할 수 있는 제어변수를 찾는 일이다. 이를 위해서 다음과 같이 앞 절의 운동방정식을 기준궤적에 대해서 선형화한 운동방정식이 Gamma 유도법칙을 유도하기 위해서 필요하게 된다. 식 (1)을 선형화하면

$$\delta \dot{x} = F \delta x + G \delta \lambda \quad (9)$$

$$\text{여기서 } F = \frac{\partial \dot{x}}{\partial x}, \quad G = \frac{\partial \dot{x}}{\partial \lambda}.$$

F와 G는 시간에 대한 함수로 기준궤적에 대해서 구해진다. 선형화된 운동방정식의 해는

$$\delta x(t) = \Phi(t, t_0) \delta x(t_0) + \Gamma_x(t, t_0) \delta \lambda \quad (10)$$

여기서  $\Phi(t, t_0)$ 는 상태전이행렬이고,  $\Gamma_x(t, t_0)$ 는 제어행렬이다. 증력벡터에서 2차 이상의 항을 무시할 경우

$$F = \begin{bmatrix} 0_{3 \times 3} & I \\ F' & 0_{3 \times 3} \end{bmatrix}, \quad F' = -\frac{\mu I}{r^3} + \frac{3\mu r r^T}{r^5} \quad (11)$$

$$\frac{\partial \dot{x}}{\partial \Delta v} = \begin{bmatrix} 0_{3 \times 3} \\ I \end{bmatrix}, \quad \frac{\partial \dot{x}}{\partial t_i} = \begin{bmatrix} -\Delta v_i \\ 0_{3 \times 1} \end{bmatrix} \quad (12)$$

여기서  $I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ 로 항등 행렬을 나타낸다.

상단 KM이 낼 수 있는 속도 임펄스의 크기는 정해져 있으므로 KM 제어변수는 다음과 같이 제한되어진다.

$$\delta |\Delta v_s|^2 = \frac{\partial |\Delta v_s|^2}{\partial \Delta v_s} \delta \Delta v_s = 2 \Delta v_s^T \delta \Delta v_s = 0 \quad (13)$$

목표궤도 달성을 위한 제한조건은 다음과 같이 상태변수의 함수로 주어진다.

$$y = f_c(\vec{x}) \quad (14)$$

이 방정식을 선형화하면

$$\delta y = H \delta x, \quad H = \frac{\partial y}{\partial x} \Big|_{x(t_F)} \quad (15)$$

위 방정식들로부터 다음과 같이 선형화된 구속조건방정식을 얻을 수 있다.

$$\delta y = H \delta x_F = H \Phi \delta x_o + H \Gamma_x \delta \lambda \quad (16)$$

$$\delta y = \Gamma \delta \lambda \quad (\Gamma \equiv H \Gamma_x, \quad \delta x_o = 0_{6 \times 1}) \quad (17)$$

### 2.3 유도 알고리즘

KM의 전체 속도 임펄스는 정해져 있기 때문에 IUS의 RCS는 KM의 임펄스 오차를 보상하는 데 필요한 속도 증분을 제공하는 역할을 한다. 유도방정식의 해는 예측하지 못한 섭동으로 인해 에너지가 부족한 조건이 유발되면 이로 인해 해가 존재하지 않을 수 있는데, 이 경우 RCS의 연소를 고려해서 유도 문제를 구성하면 유도 방정식의 해가 존재하는 조건으로 변경할 수 있다. 이 해가 실제 연소 가능한 RCS의 연료량보다 더 많은 값을 요구할 수도 있다. 여기서는 Fig. 1과 같이 발사체 상단에 속도 보상 시스템이 있고 상단 연소종료직후에 작동하는 경우에 대해 유도알고리즘을 구성해 보았다.

다음은 Fig. 1과 같은 조건에 대해 Gamma 유도방식을 적용하기 위한 무추력 구간과 추력 구간 각각에 대한 유도 알고리즘을 기술하였다.

(가) 무추력 구간에서의 유도 알고리즘

①  $t_1, \Delta v_s$ 를 추정한다.

여기서  $t_1$ 은 다음과 같이 계산할 수 있다.

$$t_1 = t_{ig} + (\Delta T_B - \Delta t_1), \quad \Delta t_1 = \frac{\iint_{t_0}^{t_{BO}} a(t) dt^2}{\Delta V_2} \quad (18)$$

여기서  $\Delta V_2$ 는 상단 구간동안 얻을 수 있는 속도를 나타낸다.  $\Delta v_s$ 는 발사 전에 목표궤도를 만족시킬 수 있는 값으로 미리 계산해서 사용한다. 즉 Gamma 유도방법을 사용하기 위해서는 사전에 계산된 제어변수 값이 필요하게 된다.

초기에 RCS를 통해 얻을 수 있는 속도  $\Delta v_R = 0$ 으로, RCS 임펄스 시간은  $t_{1R} = t_{BO}$ 로 상단 연소종료시점으로 두었다.

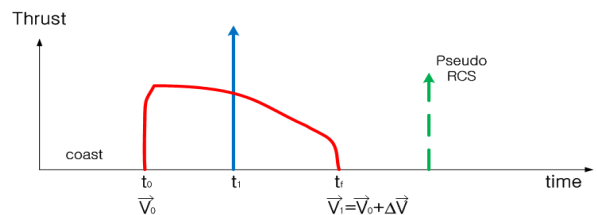


Fig. 1. Simplification of upper stage thrust profile using velocity impulse model

② 주어진  $t_1, \vec{\Delta v}_s, t_{1R}, \vec{\Delta v}_R$ 을 사용하여 비행궤적을 얻는다. 먼저  $t_1, \vec{\Delta v}_s$ 에 대해 2.1절에서 단순화된 운동방정식을 적분함으로써 연소종료시점에서의 상대벡터를 얻고 여기에  $\vec{\Delta v}_R$ 을 더해 줌으로써 RCS 영향이 고려된 최종 상태 벡터  $\vec{x}_F$ 를 얻는다. 이때  $t_{1g}$ 도  $t_1$ 으로부터 계산한다.  $t_{BO}$  시점까지  $J_2$  이상의 항을 무시할 수 있다면  $t_1$  전후의 궤적을 각각 Keplerian 궤적으로 대신함으로써 적분과정을 생략할 수 있다. 본 논문에서 사용된 발사체 모델의 경우 수치적분을 사용한 경우와 Keplerian 궤적을 사용한 경우 거의 차이가 없음을 확인하였다. 따라서 적분과정이 없도록 알고리즘을 구성하였다.

③ 목표궤도를 만족시키기 위한 제한조건과 행렬  $\Gamma$ 를 계산한다.

$$(i) \vec{y} = \vec{f}_c(\vec{x}_F)$$

발사체 상단에 다음과 같이 각운동량 크기  $|\vec{h}|$ , 각운동량 중 지구 자전축 방향 성분  $h_3$ , 이심률 벡터의 지구 자전축 방향 성분  $e_3$ , 궤도 에너지  $C_3$ 로 구성된 4개의 제한조건을 적용해보았다. 이 4개의 제한 조건은 목표 궤도의 장반경, 이심률, 경사각, 근지점 이각 조건으로도 나타낼 수 있다[1].

$$y_1 = h$$

$$y_2 = h_3$$

$$y_3 = e_3$$

$$y_4 = C_3$$

여기서

$$\vec{h} = \vec{r} \times \vec{v}$$

$$\vec{e} = \vec{v} \times \vec{h} / \mu - \vec{r} / r$$

$$C_3 = v^2 - 2\mu/r$$

$$(ii) \Gamma = H\Gamma_x = H(\vec{x}_F)\Gamma_x(t_{BO})$$

여기서  $\Gamma_x(t_{BO})$ 는 다음과 같은 미분방정식을 현재시각  $t_o$ 부터  $t_{BO}$ 까지 적분함으로써 얻어진다.

$$\dot{\Gamma}_x = F\Gamma_x + G, \quad \Gamma_x(t_o) = 0 \quad (19)$$

운동방정식을 속도 임펄스를 사용하여 단순화할 경우 행렬  $\Phi$ 의 닫힌 해(중력벡터에 대한 2차 이상의 항을 무시할 경우)를 사용하여  $\Gamma_x$ 에 대한 닫힌 해를 얻을 수 있다[7,8]. 다음 식에서  $\Gamma_x(t_{BO})^{RCS}$ 는  $t_{BO}$ 까지 적분을 통해서 얻어진  $\Gamma_x$ 에 더해주는 값이다.

$$\Gamma_x(t_{BO}) = \Phi(t_{BO}, t_1)\Gamma_x(t_1) + \Gamma_x(t_{BO})^{RCS} \quad (20)$$

여기서

$$\Gamma_x(t_1) = \begin{bmatrix} 0_{3 \times 3} & -\vec{\Delta v}_s & 0_{3 \times 3} \\ I & 0_{3 \times 1} & 0_{3 \times 3} \end{bmatrix} \quad (21)$$

$$\Gamma_x(t_{BO})^{RCS} = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 1} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 1} & I \end{bmatrix} \quad (22)$$

(마)절에서는 이 적분을 위해 참고문헌 [7,8]의 방법에 대해서 간단히 기술하였다. 구속조건을 (i)와 같이 둘 경우 행렬  $H$ 는 다음과 같은 편미분방정식을 사용하여 계산된다.

$$\frac{\partial \vec{h}}{\partial \vec{x}} = [-[\vec{v} \times] \quad [\vec{r} \times]]$$

$$\frac{\partial \vec{e}}{\partial \vec{x}} = \left[ \left( \frac{v^2}{\mu} - \frac{1}{r} \right) I - \frac{\vec{v}\vec{v}^T}{\mu} + \frac{\vec{r}\vec{r}^T}{r^3}, \quad \frac{1}{\mu} \{ 2\vec{r}\vec{v}^T - (\vec{r} \cdot \vec{v})I - \vec{v}\vec{v}^T \} \right]$$

$$\frac{\partial h}{\partial \vec{x}} = \left[ -\frac{\vec{h}^T[\vec{v} \times]}{h}, \quad \frac{\vec{h}^T[\vec{r} \times]}{h} \right]$$

$$\frac{\partial C_3}{\partial \vec{x}} = \left[ \frac{2\mu\vec{r}^T}{r^3}, \quad 2\vec{v}^T \right]$$

여기서  $[\vec{r} \times] = \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix}$ 로 반대칭 행렬을 나타낸다.

④ 다음 선형방정식의 해를 구한다.

$$\begin{bmatrix} y_D - \vec{y} \\ 0 \end{bmatrix} = \begin{bmatrix} \Gamma_{4 \times 7} & 0 \\ 2\Delta v_S^T & 0_{1 \times 3} \end{bmatrix} \begin{bmatrix} \delta \Delta v_S \\ \delta \Delta v_R \\ \delta t_1 \end{bmatrix}, \quad \delta \lambda = \begin{bmatrix} \delta \Delta v_S \\ \delta \Delta v_R \\ \delta t_1 \end{bmatrix} \quad (23)$$

⑤  $\delta \lambda$ 가 원하는 값 이하로 작아지면 반복계산을 마친다. 그렇지 않으면 새로운  $\vec{\lambda} \leftarrow \vec{\lambda} + \delta \vec{\lambda}$ 를 가지고 단계 ②로 되돌아간다.

위 과정을 그림으로 나타내면 Fig. 2와 같다[1].

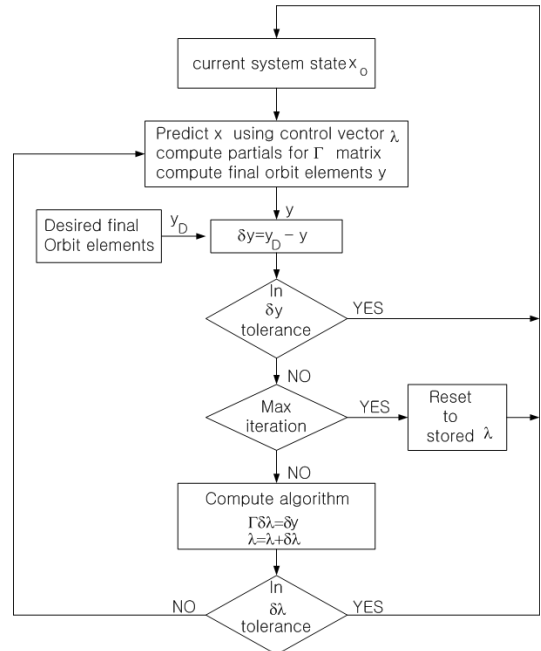


Fig. 2. Computational process of Gamma guidance algorithm

(나) 상단 연소구간에서의 유도알고리즘

제어변수에 점화시간이 포함되지 않는다는 점과 KM이 연소됨에 따라 속도 임펄스의 크기가 변한다는 점이 무추력구간과 차이가 있다.

- ① 무추력구간에서 얻어진  $\Delta \vec{v}_S$ ,  $\Delta \vec{v}_R$ 을 사용한다.
- ② 주어진  $\Delta \vec{v}_S$ ,  $\Delta \vec{v}_R$ 을 사용하여 단순화된 운동방정식을 적분(또는 Keplerian 궤적을 사용)하여  $\vec{x}_F$ 을 얻는다. 여기서 KM 가속도 프로파일을 일정가속도로 근사할 경우

$$t_1 = t_{BO} - \frac{\iint_{t_o}^{t_{BO}} a dt^2}{|\Delta \vec{v}_S|} \approx t_{BO} - \frac{\frac{1}{2} \bar{a} (t_{BO} - t_o)^2}{(\Delta V_2 - \int_{t_o}^{t_{BO}} a dt)} \quad (24)$$

$$|\Delta \vec{v}_S| = \Delta V_2 - \int_{t_o}^{t_{BO}} a dt$$

- ③ 목표궤도 구속조건 방정식과  $\Gamma$ 를 계산한다. 목표 궤도 구속조건은 무추력 구간과 동일하고  $\Gamma$ 계산에는 제어변수가 줄어들에 따라 다음과 같은 식 (25)와 (26)이 식 (20)에 적용되었다.

$$\Gamma_x(t_1) = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} \\ I & 0_{3 \times 3} \end{bmatrix} \quad (25)$$

$$\Gamma_x(t_{BO})^{RCS} = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I \end{bmatrix} \quad (26)$$

- ④ 다음 선형 방정식의 해를 구한다.

$$\begin{bmatrix} \delta \vec{y} \\ 0 \end{bmatrix} = \begin{bmatrix} \Gamma_{3 \times 6} \\ 2 \Delta \vec{v}_S^T & 0_{1 \times 3} \end{bmatrix} \begin{bmatrix} \delta \Delta \vec{v}_S \\ \delta \Delta \vec{v}_R \end{bmatrix} \quad (27)$$

여기서 시간이 흐름에 따라  $\Delta \vec{v}_S$ 의 크기도 변한다.

- ⑤ (가)절의 무추력 구간 알고리즘 ⑤와 동일  
(가)절과 (나)절에서 구속조건과 제어변수를 요약하면 Table 1, 2와 같다.

Table 1. Constraints of Gamma guidance

No.	Constraints	Coast	Burn
1	Angular momentum magnitude $h$	0	0
2	Earth rotational axis component of angular momentum $h_3$	0	0
3	Earth rotational axis component of eccentricity vector $e_3$	0	0
4	Orbital energy $C_3$	0	0
5	KM velocity impulse magnitude $ \Delta \vec{v}_S $	0	0
Total		5	5

Table 2. Control variables of Gamma guidance

No.	Control variables	Coast	Burn
1	KM velocity impulse x	0	0
2	KM velocity impulse y	0	0
3	KM velocity impulse z	0	0
4	KM ignition time	0	x
5	RCS velocity impulse x	0	0
6	RCS velocity impulse y	0	0
7	RCS velocity impulse z	0	0
Total		7	6

(다) 유도 알고리즘의 종료

KM 연소종료시점으로 접근함에 따라 유도명령은 작은 섭동에도 점점 민감해지게 된다. 이런 민감성은 작은 위치와 속도 오차에 대해서 큰 추력벡터방향 변화를 야기하고, 제어시스템의 한계를 초과하는 큰 자세 변화율을 요구할 수 있다. 이런 문제를 방지하기 위해선 예측된 연소종료시점 이전의 일정 시간부터 유도명령을 일정한 값으로 유지해야 한다.

(라) 선형 방정식의 해

앞의 알고리즘에서 알 수 있듯이 구속조건의 수(m)와 제어변수의 수(n, 미지수)에 따라 주어진 선형방정식이 정확한 해를 갖는 경우(m=n, full rank)는 해를 쉽게 구할 수 있다. 하지만 구속조건의 수가 제어변수보다 많거나(m>n) 적은 경우(m<n)도 발생하게 된다. 이럴 경우에도 적합한 유도명령은 필요하게 된다. Singular Value Decomposition(SVD) 방법의 경우 과결정(overdetermined) 또는 과소결정(undetermined) 연립방정식의 경우에 가장 작은 크기의 해를 구해준다는 장점이 있다. 즉 이전 단계에서의 제어변수에서 현재의 구속조건을 가능한 많이 만족시키면서 그중에서 보정되는 양이 가장 작은 벡터를 찾아준다. 또한 SVD 방법을 사용하면 주어진 구속조건에 의해서 얻어진 mxn 행렬이 특이값이 가까운 경우에도 LU decomposition이나 Gaussian elimination 방법보다 해를 더 잘 찾아준다는 장점이 있다[9].

(마) Keplerian 궤적에 대한 상태전이행렬 [7,8]

Keplerian 문제는 시간  $t_o$ 에서 주어진 위치  $\vec{r}(t_o) = \vec{r}_o$ 와 속도  $\vec{v}(t_o) = \vec{v}_o$ 에 대해서 시간  $t$ 에서 다음 방정식을 만족시키는 위치  $\vec{r}(t)$ 와 속도  $\vec{v}(t)$ 를 찾는 문제다.

$$\frac{d^2 \vec{r}}{dt^2} = -\frac{\mu}{r^3} \vec{r} \quad (28)$$

이 문제의 해는 다음과 같은 형태로 나타낼 수 있는데

$$\begin{bmatrix} \vec{r} \\ \vec{v} \end{bmatrix} = \begin{bmatrix} fI & gI \\ \dot{f}I & \dot{g}I \end{bmatrix} \begin{bmatrix} \vec{r}_o \\ \vec{v}_o \end{bmatrix} \quad (29)$$

여기서  $f, g, \dot{f}, \dot{g}$ 는  $t, \vec{r}_o, \vec{v}_o$ 에 대한 함수이다. 상태전이 행렬은 다음과 같이 정의되고

$$\Phi(t, t_o) = \begin{bmatrix} \vec{R}(t) & \vec{R}(t) \\ \vec{V}(t) & \vec{V}(t) \end{bmatrix} = \begin{bmatrix} \frac{\partial \vec{r}}{\partial \vec{r}_o} & \frac{\partial \vec{r}}{\partial \vec{v}_o} \\ \frac{\partial \vec{v}}{\partial \vec{r}_o} & \frac{\partial \vec{v}}{\partial \vec{v}_o} \end{bmatrix} \quad (30)$$

위 두 식으로부터

$$\begin{bmatrix} \frac{\partial \vec{r}}{\partial \vec{r}_o} \\ \frac{\partial \vec{v}}{\partial \vec{r}_o} \end{bmatrix} = \Phi(t, t_o) \begin{bmatrix} \vec{r}_o \\ \vec{v}_o \end{bmatrix} = \begin{bmatrix} \delta f I & \delta g I \\ \delta \dot{f} I & \delta \dot{g} I \end{bmatrix} \begin{bmatrix} \vec{r}_o \\ \vec{v}_o \end{bmatrix} + \begin{bmatrix} f I & g I \\ \dot{f} I & \dot{g} I \end{bmatrix} \begin{bmatrix} \frac{\partial \vec{r}_o}{\partial \vec{r}_o} \\ \frac{\partial \vec{v}_o}{\partial \vec{v}_o} \end{bmatrix} \quad (31)$$

앞의 항을  $\delta \vec{r}_o, \delta \vec{v}_o$ 로 나타내면 상태전이 행렬이 구해진다. 이 과정은 참고문헌 [7,8]에 자세히 기술되어 있다.

### III. 유도 알고리즘 성능 분석

앞 절에서 기술한 유도 알고리즘을 3-DOF 시뮬레이션에 적용하여 궤도투입 성능을 살펴보았다. 위성 발사체의 목표 궤도는 나로호와 같이 근지점 고도 300 km, 원지점 고도 1,500 km의 타원 궤도로 목표 경사각은 나로우주센터에서 요기동 없이 방위각 170 (deg)로 발사할 경우 얻어지는 경사각으로 약 80 (deg)로 설정하였다. 나로호의 최종단에는 속도 보상을 위한 RCS가 없었으나, 여기서는 IUS와 같이 속도 오차 보상 시스템이 있다고 두었다. 이전 단계에서 누적된 오차를 최종단에서 보상하여 최종단 연소 종료 후 궤도 투입되는 시나리오를 다루었다.

Gamma 유도법칙에 사용된 파라미터 값들은 다음과 같다. 유도 명령 계산은 매 10 (sec)마다 수행했으며, 반복 계산 횟수는 5회로 제한하였다. 엔진연소종료 시점에 가까워지면 작은 투입 오차라도 이를 줄이기 위한 유도 명령은 커질 수 있어 엔진연소종료 10 (sec) 전에 유도 명령 계산을 중단하였고, 이후 유도 명령은 이전까지 계산된 유도 명령을 유지하였다.

Figures 3, 4는 매 10 (sec)마다 반복 계산을 통해 궤도 투입 구속조건 파라미터 중  $C_3$  오차와 KM 속도 x-축 성분의 업데이트량 변화를 나타낸다. 매 계산주기마다 빠르게 수렴함을 알 수 있다. 무추력 구간에서 유도 시작 직후, 최종단 점화 직후에 오차가 크지만 줄어들음을 알 수 있다.

궤적 투입 정밀도 분석을 위해 이전 단계에서 누적된 다양한 위치 및 속도 오차에 대한 3,000번의 Monte Carlo 시뮬레이션을 수행하였다. 최종단에서 개루프 명령을 적용하는 경우 이런 오차 요인은 Table 3과 같은 궤도 투입 오차를 발생시킨다. 여기서 R(PER)은 근지점 고도 오차를, R(APO)는 원지점 고도 오차를, INC는 경사각 오차를, AOP는 근지점 이각 오차를 나타낸다. 얻어진 궤도 투입 오차는 오차 분포의 최대값을 나타낸다. 반면에 Gamma 유도 방식을 적용하면 설정한 궤도 투입 구속 조건을 만족시키도록 유도하기 때문에 궤도 투입 오차가

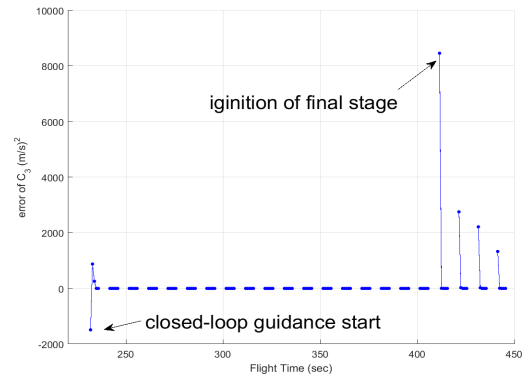


Fig. 3. Error of C3 variation during guidance computation cycle

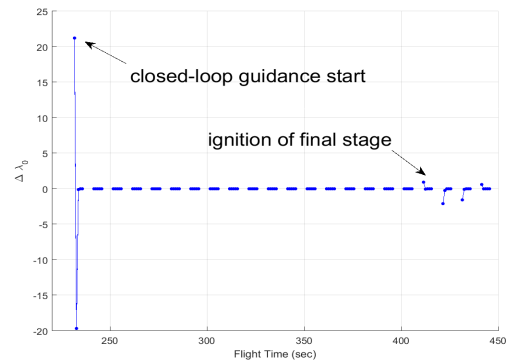


Fig. 4. Update of KM x-axis velocity impulse during guidance computation cycle

Table 3. Orbit injection accuracy

Guidance method	R(PER) (km)	R(APO) (km)	INC (deg)	AOP (deg)
Open-loop	1.23	30.24	0.042	0.401
Gamma (1)	0.24	1.39	0.002	0.293
Gamma (2)	0.24	1.38	0.002	0.293

줄어들게 됨을 알 수 있다. Gamma (1)은 Kerplerian 궤적을 적용한 경우로 수치적 적분 방법을 사용한 경우 Gamma (2)와 거의 차이가 없음을 알 수 있다.

Figures 5, 6은 3,000개 시뮬레이션 결과 중 30가지 경우에 대한 최종단 연소 구간 중 유도 명령을 나타낸다. 유도 계산 주기 동안 얻어진 유도 명령은 유지되고, 10 (sec) 후 새로운 유도 명령이 계산되면 유도 명령은 변하게 됨을 알 수 있다. 유도 명령이 유지되는 동안은 제어시스템에 부담이 적으나 유도 명령이 변하게 되면 제어시스템을 통해 얻어지는 구동기 명령도 이를 추종하기 위해 명령 변화가 요구된다. 주어진 위성발사체의 비행 조건 중 큰 명령 변화가 발생하는 조건을 고려해야 되면 Gamma 유도 방식의 경우 매 유도 주기마다 발생하는 유도 명령 변화를 줄여줄 수 있는 유도 필터가 필요할 것으로 보인다.

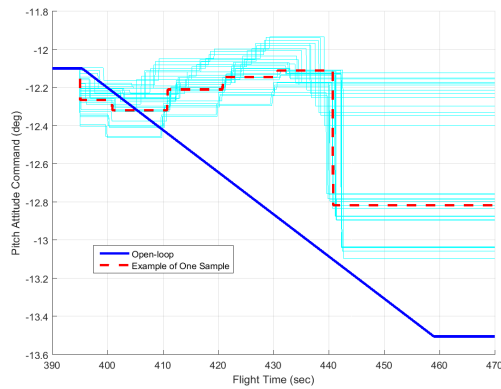


Fig. 5. Pitch attitude commands

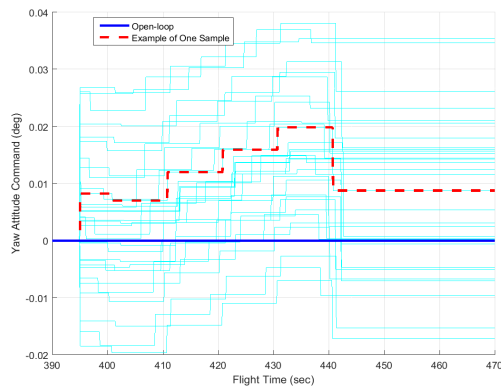


Fig. 6. Yaw attitude commands

개루프 명령을 적용하는 경우 최종단의 점화 시간은 궤도 투입 오차 보상을 위해 변동되지 않는다. 반면에 페루프 유도 방식을 적용하면 본 논문의 Gamma 유도 방식을 적용한 경우와 같이 점화 시간이 변하게 된다. Fig. 7은 3,000번의 시뮬레이션을 통해 얻어진 점화 시간 분포를, Fig. 8은 RCS 속도 증분 분포를 나타낸다. 위성발사체의 최종단에 RCS를 장착할 경우 RCS 속도 증분으로부터 RCS 요구 추진제를 계산할 수 있게 된다. 예를 들어 이전 단계까지 누적된 오차를 99.73% 확률로 보상이 필요한 경우에는 Fig. 8로부터  $\Delta V = 47.54$  (m/s)에 해당

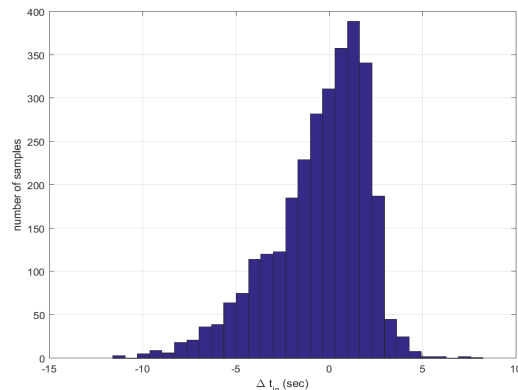


Fig. 7. Ignition time variations of the final stage

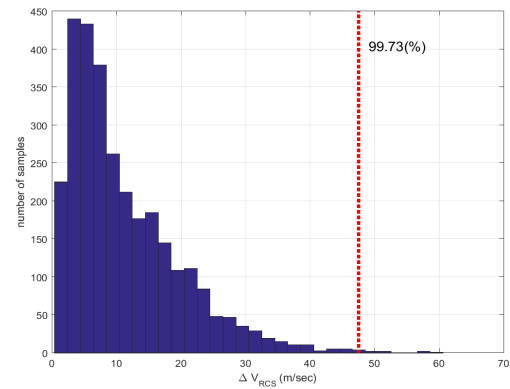


Fig. 8. Velocity to be gained using RCS

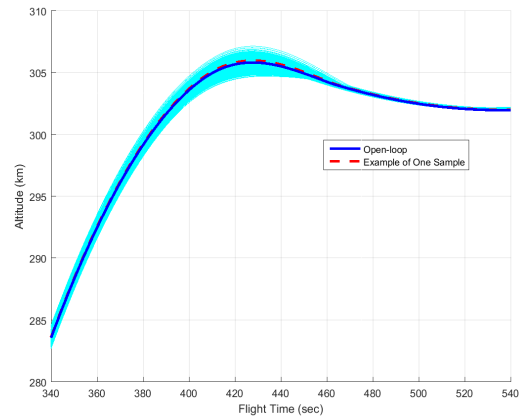


Fig. 9. Altitude : Gamma guidance

하는 추진제가 요구된다. Fig. 9는 Gamma 유도 방식에 의해 상단 연소 종료 후 목표 궤도 고도에 투입되는 궤적을 나타낸다.

#### IV. 결 론

본 논문에서는 IUS에 사용된 유도알고리즘을 주어진 위성발사체의 최상단 고체모터 유도를 위해 적용해 보았다. 알고리즘의 수렴성을 위해 고체모터의 속도 오차 보상시스템이 있는 경우로 가정했으며, 알고리즘 단계 중 최종상태변수 추정을 위한 적분과정을 Keplerian 궤적으로 대신함으로써 알고리즘을 단순화시켰다. 최상단 이전 단계까지 누적된 궤적 오차가 있는 조건에 대해 개루프 명령을 적용한 경우와 3-DOF 시뮬레이션을 수행해 비교함으로써 유도법칙의 성능을 살펴보았다. 시뮬레이션 결과 속도 오차 보상 시스템과 Gamma 유도 알고리즘을 사용하여 고체모터의 단점을 보완할 수 있음을 확인할 수 있었다.

알고리즘의 특성과 이상의 결과로 볼 때 이 유도방식은 일정유도명령을 출력하기 때문에 제어시스템에 부담을 덜어줄 수 있고, 반복 계산 시작 시 초기 유도명령과 상단 점화시간 정보가 사전에 필요함을 알 수 있었다.

추후에는 목표 궤도 구속조건 대비 제어 변수가 많은 조건에 대해 RCS 추진제 사용량을 가능한 줄여주는 해를 사용하는 경우 유도 성능과 알고리즘 계산량 증가에 대한 분석을 수행할 예정이다.

## References

- 1) Hardtla, J. W., "Gamma Guidance for the Inertial Upper Stage (IUS)," *Guidance and Control Conference*, August 1978, pp. 357~362.
- 2) Hardtla, J. W. and Schultz, L. A., "Inertial Upper Stage /IUS/ Multi-mission Adaptability via Gamma Guidance Software Menu Tables," *Guidance and Control Conference*, 30 January 1982, pp. 213~227.
- 3) Kuhns, R. H. and Coomer, G. C., "Gamma Guidance - Design and Results for two Flights," *Proceedings of the Seventh Annual Rocky Mountain Conference*, February 1984, pp. 295~310.
- 4) Dasgupta, S. and Ramakrishna, S., "Explicit VG Guidance Algorithm for a Solid Powered Closed Loop Guidance Mission," *IFAC Proceedings*, June 1985, pp. 189~195.
- 5) Yokota, K., et al., "Development and Flight Results of Guidance and Control System for 4th Epsilon Launch Vehicle," *30<sup>th</sup> ISTS*, June 2019.
- 6) Song, E. J. and Roh, W. R., "Modified Gamma Guidance for Satellite Launch Vehicles," *Korea Automatic Control Conference*, October 2005, pp. 492~498.
- 7) Battin, R. H., *An Introduction to the Mathematics and Methods of Astrodynamics*, AIAA Education Series, Revised Edition, 1999, pp. 128~131.
- 8) Der, G. J., "An Elegant State Transition Matrix," *The Journal of the Astronautical Sciences*, Vol. 45, No. 4, 1997, pp. 371~390.
- 9) Press, W. H., et al., *Numerical Recipes in C : The Art of Scientific Computing*, Cambridge University Press, Vol. XXVII, No. 1, 1979, pp. 1~38.