

# 선형 탄젠트 피치 프로그램을 이용하는 Implicit 유도

진재현      탁민계

한국과학기술원 항공우주공학과

## Implicit Guidance Using Linear Tangent Pitch Program

Jin, Jaehyun      Tahk, Min-Jea

Dept. of Aerospace Eng., KAIST

**ABSTRACT** Implicit guidance algorithm can deal with the trajectory error rapidly, but it has to save much data. If, however, the control variable is represented by a specific function form, a few parameters will suffice to define the control variable. In this paper, we study the method of updating the parameters of the control function for the reduction of trajectory errors. The method proposed here does not require much memory for guidance.

### 1. 서론

우주 비행체를 유도하는 알고리즘은 크게 둘로 나뉜다. implicit 식과 explicit 식의 유도 방법이 된다. 본 논문에서는 이 두 방법의 일반적인 내용과 함께 단순화된 implicit 방식을 소개한다.

전자는 delta 혹은 perturbation 유도라고 불리기도 한다. 이에 해당하는 것으로, 일반적으로 많이 사용되는 것이 neighboring optimum guidance<sup>[1]</sup> 라고 알려진 것이다. 이 방식은 지금의 상태에서 목표지점에 도달하기 위해 필요한 제어량을 계산하는 것이 아니라, 궤적 최적화 작업을 통해서 구한 최적 궤적을 이용한다. 즉, 이미 목표지점에 도달하는 최적 궤적을 알기 때문에 그 최적 궤적을 기준 궤적이라고 생각하고 실제궤적과 기준 궤적을 비교하면서 그 차이를 보상해주는 방법이다. 이 방법을 위해서는 시간에 따른 기준 궤적의 데이터가 저장되어 있어야 한다. 그리고 각 시각(유도 명령을 계산하는 시각)에서 궤적과 제어 변수의 미소변화가 최종점 구속조건에 미치는 영향에 대한 민감도의 데이터를 모두 가지고 있어야 한다. 이러한 데이터는 발사전에 기준 궤적으로부터 계산할 수 있다. 그래서 유도 명령을 자주 개선하여 보다 정확하게 비행체를 유도할 수 있지만 결과적으로 많은 데이터의 기억을 필요로 한다. 그러나 만약 발사기지에서 유도 명령을 내리는 방법을 이용하면 기억용량이 크게 문제되지는 않는다. 이 방식은 기준 궤적으로부터 작은 양의 변동을 가정하므로 오차 궤적의 방정식은 선형화된 형태가 된다. 유도 명령(기준 제어

변수의 수정량)은 궤적오차에 미리 계산해 놓은 이득을 곱해서 구해지기 때문에 계산량은 많지 않으며 따라서 오차의 처리가 빠르다. 처리속도가 빠르기 때문에 유도명령 개선 간격을 줄일 수 있고 결과적으로 보다 정확한 정밀도를 유지할 수 있게 된다. 그러나 작은 양의 변동을 가정했기 때문에, 큰 오차가 발생할 경우 유도 성능은 떨어진다. (참고문헌 [1], [2])

Explicit 방식은 현재 상태에서 목표지점에 도달할 수 있는 궤적을 그때 그때 계산하는 것이다. 그러나 일반적으로 궤적 최적화와 같은 비선형 이점 경계치(two-point boundary value problem)의 문제를 푸는데는 많은 시간이 걸린다. 그래서 유도 문제에서는 실시간 계산을 위해서 가급적 근사화된 식을 이용한다. 그렇지만 연속적인 유도를 함으로서 근사화의 오차를 보상하게 되고 최종점 구속조건을 만족시킬 수 있다. 그리고 임무의 다양성을 계획할 수 있고 또한 많은 데이터를 저장할 수 있는 기억장치가 필요없다. 참고문헌 [3], [4], [5]에서는 지구 근처의 저고도에 진입하는 발사체의 explicit 유도를 다루고 있는데 제어 변수(여기서는 피치 프로그램)를 먼저 선형 탄젠트 법칙이라는 형태로 가정하고 현재의 상태와 최종점의 상태를 비교해서 계속해서 선형 탄젠트 법칙의 파라미터를 개선하는 방법을 이용하고 있다.

본 논문에서는 implicit 방식을 이용하는 유도에 관한 내용을 다루고 있다. 앞에서 언급한 것처럼 implicit 방식은 많은 데이터의 저장을 필요로 하지만 본 논문에서는 이런 문제점을 피할 수 있는 방법을 제시한다. 제어 법칙을 특정한 함수의 형태로 가정하는 방법, 즉, 파라미터화 된 제어 법칙을 이용한다. 이 파라미터화 된 제어 법칙은 단지 몇 개의 파라미터를 정하면 전 시간영역에서 필요한 제어 변수값을 알 수 있으므로 데이터의 양이 문제가 되지 않는다. 파라미터화 된 제어법칙을 이용해서 비행체를 제어할 경우에 대해서 궤적오차가 생기면 파라미터를 적절히 수정하여 최종적으로 목표지점에 도달할 수 있게 한다. 이 방법의 적용 예제로서 인공위성을 지구 저고도에 진입시키는 발사체의 문제와 단순화된 후성간 궤도선이 문제를 택해서 각각 전산 모의 시험을 수행했다.

## 2. 궤적 최적화 기법

기존 궤적으로 이용하게 될 최적 궤적을 먼저 구해야 한다. 본 장에서는 두가지 방법을 간략하게 소개한다.

### 2.1 최적 제어 이론

다음과 같은 최적화의 문제를 살펴보자.

$$\text{minimize } J = \phi(x_f, t_f) + \int L(x, u, t) dt \quad (1)$$

$$\text{subject to } \ddot{x}(x_f, t_f) = 0 \quad (2)$$

$$\dot{x} = f(x, u, t), \quad x(t_0) \text{ given} \quad (3)$$

$x(t)$  는 비행체의 위치와 속도를 나타내는 상태 벡터이고  $u(t)$  는 그 상태 벡터를 조정할 수 있는 제어 변수인데, 보통은 추력벡터의 방향이 된다.  $J$  는 최소화 하고자 하는 성능함수인데 비행도중 그리고 최종점에서 계산된다. 그리고 구속조건은 각각 최종점 구속조건과 동력학적 구속조건이 된다.

위의 문제는 변분법(calculation of variation)을 이용해서 최적 궤적이 만족해야 할 필요조건을 구하면, 그 해는 Euler-Lagrange 방정식의 형태로 나타난다. 즉, 새로운 변수 Lagrange multiplier  $\lambda$  를 도입해서 Hamiltonian  $H$  를 정의한다.

$$H = L + \lambda^T f \quad (4)$$

$$\dot{\lambda} = -(\partial H / \partial x) \quad (5)$$

$$\lambda(t_f) = \partial \phi / \partial x |_{t_f} \quad (6)$$

$$\ddot{\phi} = (\phi + \lambda^T \ddot{x}) \quad (7)$$

그러면 필요 조건은 다음과 같다.

$$0 = Q = [ d\phi/dt + L ] |_{t_f} \quad (8)$$

$$0 = (\partial H / \partial u) \quad (9)$$

이 결과식을 이점 경계치 문제(two point boundary value problem)라고 하는데 이 문제를 해결하기 위한 많은 수치적 방법이 있다.(참고문헌 [6], [7]) 결과적으로 최적 궤적을 만드는 제어 변수의 시간에 따른 값이 구해진다.

### 2.2 비선형 프로그래밍

최적 제어 이론에서 제시하는 해는 최적 궤적이 만족해야 하는 필요 조건이다. 그러나 만약 제어 변수의 함수를 특별한 형태로 가정하고 그 함수의 파라미터를 독립변수로 생각한다면, 파라미터의 값을 적절하게 조정하여 성능함수의 극값을 찾을 수 있다.

제어 변수 함수의 파라미터를 독립변수로 정한다. 위에서

설명하겠지만 이 함수는 다항식 혹은 많이 사용되는 피치 프로그램의 형태를 택한다.

$$u(t), \quad t_0 \leq t \leq t_f \quad (10)$$

$$\rightarrow u(t) = U(p, t), \quad t_0 \leq t \leq t_f, \quad p \in R^n \quad (11)$$

$p$ 는 파라미터들로 이루어진 벡터이고 그 파라미터의 수는 임의의  $n$ 개로 가정한다.

성능함수는 독립변수를 통해서 조정되는 양을 택한다. 보통은 Bolza 형으로 나타낸다. 성능함수  $J$  를 최소화하기 위해 파라미터벡터  $p$  가 만족해야 할 필요조건은 다음과 같다. 두 개의 구속조건(식 (2)와 (3))을 만족시키면서 극값조건을 만족시켜야 한다.

$$0 = \partial J / \partial p \quad (12)$$

이 파라미터벡터를 찾는 수치해석 기법은 참고문헌 [8], [9], [10] 에서 찾아볼 수 있다. 2.1절의 방법과 비교해볼 때, 이 방법을 이용하면 함수의 파라미터 몇개만 기억하고 있으면 전 시간 영역에서 필요로 하는 제어 변수 값을 알 수 있다. 그래서 이 방법은 훨씬 적은 양의 기억장치를 필요로 한다.

### 2.3 제어 변수의 함수화

참고문헌 [9] 는 단순화된 특성간의 궤도 천이 문제를 다루고 있는데, 여기서는 추력벡터의 방향을 제어 변수로 생각하고 시간에 대한 4차 식으로 표현 하였다. 그리고 시간은 정규화(normalized)된 변수로 나타내었다. 즉,

$$u(\Gamma) = p_0 + p_1 \Gamma + p_2 \Gamma^2 + p_3 \Gamma^3 + p_4 \Gamma^4, \quad \Gamma = t/t_f \quad (13)$$

성능함수를 개선하기 위해서는 항을 더 추가하거나 전 시간영역을 둘로 나누어 각각에 대한 다항식을 고려해줄 수도 있다. 본 논문에서는 전 시간 영역에 대해서 4차 다항식으로 가정한다. 그러면 결정해야할 파라미터는 다음과 같다.

$$p = [p_0, p_1, p_2, p_3, p_4]^T \quad (14)$$

인공위성을 궤도에 진입시키는 발사체 문제의 경우, 발사체의 운동을 단순화시켰을때 피치각이 제어 변수가 된다. 그리고 피치 프로그램을 단순화하는 방법으로 널리 알려진 선형 탄젠트 법칙이 많이 이용된다. 일정한 시간 영역에서 이 법칙은 다음처럼 표시된다.

$$\tan u(t) = a t + b, \quad t_0 \leq t \leq t_f \quad (15)$$

이 식을 더욱 단순화하면[3],

$$u(t) = a t + b, \quad t_0 \leq t \leq t_f \quad (16)$$

처럼 쓸 수 있다. 한 시간 영역에 대해서 결정해 주어야 할 파라미터는 초기 피치값과 피치 각속도가 된다. 전체 발사시간에 대해서 일정한 피치 각속도를 가지는 시간 영역이 많으면 성능은 좋아진다. 본 논문에서 참조할 발사체는 지구 저고도 진입용으로 개발된 4 단 발사체인데 추진계는 모두 고체연료다. 이 발사체에 대한 제한은 참고문헌 [11]에 자세히 나와있다. 먼저 각 단에서 피치 각속도가 일정하다고 생각하고 부스터 구간을 첨가한다. 즉, 시간 영역을 5 개로 나눈다. 2 단 초기의 피치각은 1 단의 최종값과 같다고 둔다. 그리고 3 단과 4 단의 피치각은 고정 되었다고 생각한다. 그러면 결정해야 할 파라미터의 수는 모두 7 개가 된다.

$$p = [a_0, b_0, a_1, a_2, b_2, b_3, b_4]^T \quad (17)$$

그리고 본 논문에서 고려하는 발사체의 최적 궤적은 주어진 발사체로 최대의 유상 하중을 목표궤도에 진입시키는 궤적이다. 이 궤적을 구하는 방법은 참고문헌 [12] 에 자세히 나와있다.

### 3. 유도 알고리즘

#### 3.1 이차 변동을 이용하는 유도(neighboring optimum guidance)

비행체가 기준 궤적을 따라가다가 외란등의 요인에 의해 그 궤적에서 벗어날 경우, 이 변동이 최종적으로 성능함수에 미치는 영향을 고려하여 그 성능함수 오차를 최소화하는 방향으로 유도 알고리즘을 수행한다. 변동이 작다고 가정한다면, 성능함수의 일차 변동은 영이되고 따라서 오차를 줄이는데 아무런 도움이 되지 못한다. 그래서 이차 변동을 생각하게 되고 결과적으로 LQ(linear quadratic) 문제와 비슷한 형태의 문제로 전개된다.

성능함수 J (식 (1)) 의 이차 변동항은 다음처럼 전개된다.

$$\begin{aligned} \delta^2 J = & \frac{1}{2} [ \delta x^T ( \phi_{xx} + (\nu^T \psi_x)_x ) \delta x ] |_{t_f} \\ & + \frac{1}{2} \int_{t_0}^{t_f} [ \delta x^T \delta u^T ] \begin{bmatrix} H_{xx} & H_{xu} \\ H_{ux} & H_{uu} \end{bmatrix} \begin{bmatrix} \delta x \\ \delta u \end{bmatrix} dt \end{aligned} \quad (18)$$

그리고 최종점 구속 조건은 변하지 않는다고 가정하면 각 시각에서 상태벡터의 변동을  $\delta x(t)$  라고 할때, 제어 변수의 수정량은 다음과 같다. ( 참고문헌 [6] )

$$\delta u(t) = - (H_{uu})^{-1} [ H_{ux} + f_u^T (S - RQ^{-1}R^T) ] \delta x(t) \quad (19)$$

그리고 전 시간 영역에서 저장해두어야 할 행렬 Q(t), R(t),

S(t) 는 다음과 같은 방정식을 만족시켜야 한다.

$$\begin{aligned} dS/dt = & - SA - ATS + SBS - C, & S(t_f) = & [ \bar{\psi}_{xx} ] |_{t_f} \\ dR/dt = & - ( A^T - SB ) R, & R(t_f) = & [ \bar{\psi}_x^T ] |_{t_f} \\ dQ/dt = & R^T B R, & Q(t_f) = & 0 \end{aligned} \quad (20), (21), (22)$$

$$A(t) = f_x - f_u (H_{uu})^{-1} H_{ux} \quad (23)$$

$$B(t) = f_u (H_{uu})^{-1} f_u^T \quad (24)$$

$$C(t) = H_{xx} - H_{xu} (H_{uu})^{-1} H_{ux} \quad (25)$$

상태의 변동에 따른 최종시간의 변화도 고려해 줄 수 있다.

식(22) 에서 알 수 있듯이 최종시간에 가까워질수록 Q(t) 는 영으로 수렴하게된다. 반대로 Q<sup>-1</sup>(t) 는 큰 값으로 발산하는 현상이 생긴다. 이러한 경우에 그 값을 그대로 식(19)에 적용하면 실제 물리적인 의미가 없는 결과가 나타난다. 그래서  $\delta u(t)$  값 자체에 제한을 두어야 한다.[2]

#### 3.2 파라미터값 수정을 이용한 유도

파라미터화 된 제어 법칙을 이용할 경우에도 마찬가지로, 실제 비행시 궤적 오차가 발생한다면 그 파라미터를 적절히 수정해주어야 한다. 그 개념은 3.1 절의 이차 변동항을 고려해서 그 오차를 최소화 하는 방법을 쓴다.

최종시간이 고정되지 않은 경우에 대해 생각해보자. 최종점에서의 성능함수의 이차 변동항을 다음처럼 나타낼 수 있다.

$$J = \bar{\psi} = \phi + \nu^T \bar{\psi} \quad (26)$$

$$\delta^2 J = \frac{1}{2} [ \delta p^T \delta t_f ] \begin{bmatrix} \bar{\psi}_{pp} & \bar{\psi}_{pt} \\ \bar{\psi}_{tp} & \bar{\psi}_{tt} \end{bmatrix} \begin{bmatrix} \delta p \\ \delta t_f \end{bmatrix} \quad (27)$$

그리고 최종점 구속조건의 변동은

$$\delta \bar{\psi} - R \delta x_o = [ \bar{\psi}_p \quad \bar{\psi}_t ] \begin{bmatrix} \delta p \\ \delta t_f \end{bmatrix} \quad (28)$$

이 된다.

변수를 다음처럼 바꾼다.

$$\begin{aligned} \delta^2 J \Rightarrow F, & \quad \begin{bmatrix} \delta p \\ \delta t_f \end{bmatrix} \Rightarrow P, & \quad \begin{bmatrix} \bar{\psi}_{pp} & \bar{\psi}_{pt} \\ \bar{\psi}_{tp} & \bar{\psi}_{tt} \end{bmatrix} \Rightarrow G \\ -R \delta x_o \Rightarrow C, & \quad [ \bar{\psi}_p \quad \bar{\psi}_t ] \Rightarrow A \end{aligned} \quad (29)$$

그러면 위의 문제는 다음과 같이 표현된다.

$$\text{minimize} \quad F = \frac{1}{2} P^T G P \quad (30)$$

$$\text{subject to} \quad C = A P \quad (31)$$

즉, 선형 구속조건을 가지는 이차 성능함수(quadratic objective function)의 최소값을 찾는 문제가 된다. 그리고 이 문제는 multiplier 를 도입하여 쉽게 해를 구할 수 있다.

$$F' = \frac{1}{2} P^T G P + \lambda^T (A P - C) \quad (32)$$

$$\lambda = -[AG^{-1}A^T]^{-1}C \quad (33)$$

$$P = G^{-1}A^T[AG^{-1}A^T]^{-1}C \quad (34)$$

마찬가지로, 비행전에 기준 궤적을 이용해서 R, G, A 를 먼저 구해놓는다. 그리고 상태벡터의 오차가 생기면 C 를 계산하여 파라미터 수정량 P 를 구하게 된다. 초기 시간이 아닐때 변동이 생기는 경우나, 혹은 중간에 다시 수정량을 구하고 싶은 경우를 대비해서 적당한 시간 간격을 두고 각 시각에서 R, G, A 를 구해놓을 수도 있다. 그러나 neighboring optimum guidance 와 마찬가지로 한번 정해진 기준 궤적을 이용하는 것이므로 유도 알고리즘은 개회로(open loop)가 된다. 이 방법은 식(34)에서 알 수 있듯이 단지 행렬의 곱만을 포함하기 때문에 유도 알고리즘은 매우 간단해진다.

만일 고정된 최종시간을 가지는 문제인 경우에는 P, G, A 에서 시간과 관련된 항을 제외하면 된다.

#### 4. 문제 정의

##### 4.1 인공위성 궤도 진입용 우주 발사체 문제

식(3) 에서 사용되는 상태방정식은 Newton 의 운동법칙을 만족하는 운동방정식이 된다. 그리고 발사체를 질점으로 생각한다.

$$\ddot{R} = -\frac{\mu}{r^3} R + \frac{1}{m} F_{total} \quad (35)$$

여기서 R 은 관성좌표계에서의 위치벡터,  $r = |R|$ ,  $\mu$  는 중력계수이고  $F_{total}$  은 추력과 공기력의 합력이다. 추력 벡터는 발사체의 축과 일치한다고 가정하고 피치각을 제어변수로 정의한다. 발사체의 제원과 공력계수 그리고 발사작업에 관한 내용은 참고문헌 [11](pp.311-354) 을 참조하기 바란다.

궤도를 유일하게 결정하는 요소들은 이심률 e, 진변위각  $\theta$ , 각운동량 h 인데, 이들은 연소완료때의 발사체 상태에 의해 다음처럼 결정되어 진다.

$$h = |R \times V| \quad (36)$$

$$\tan \theta = \frac{(rv^2/\mu) \sin\beta \cos\beta}{(rv^2/\mu) \cos^2\beta - 1} \quad (37)$$

$$e^2 = \left[ \frac{rv^2}{\mu} - 1 \right] \cos^2\beta + \sin^2\beta \quad (38)$$

여기서 R, V 는 각각 위치와 속도 벡터이다. 그리고  $\beta$  는 경로각이다. 이 식에서 알 수 있듯이 연소끝점에서 속도, 고도, 그리고 경로각 세가지로써 모든 궤도를 기술할 수 있으므로 이 세가지의 조건이 식 (2)의 최종점 구속조건으로 사용된다. 그리고 최종점 구속조건은 벌칙함수의 형태로 성능함수에 첨가시켰다.

본 논문에서는 최종시간이 고정된 것으로 생각한다. 즉, 사용가능한 모든 추진제를 태우게 된다. 그래서 성능함수를 최대유상 하중으로 생각한다.

$$\text{minimize } J = W_{PL} \quad (39)$$

##### 4.2 단순화된 궤도 천이 문제

태양을 중심으로 원운동을 하고 있는 지구에서 화성궤도로 옮겨가는 문제이다. 여기에서는 단순화된 방정식을 사용한다. 즉, 비행체에 작용하는 힘은 추력과 태양에 의한 인력만 고려한다. 구좌표계로 나타낸 운동방정식은 다음과 같다.

$$du/dt = v^2/r - \mu/r^2 + T \sin\beta/m \quad (40)$$

$$dv/dt = -uv/r + T \cos\beta/m \quad (41)$$

$$dr/dt = u \quad (42)$$

여기서 u, v 는 각각 반경, 접선방향 속도이고 r 은 반경이다.  $\beta$  는 제어변수로서 추력벡터의 방향이 된다. 필요한 데이터는 다음과 같다. 정극화된 데이터들이다.

$$\text{질량} : m(t) = 1.0 - 0.0749 t$$

$$\text{추력} : T = 0.1405$$

$$\text{시간단위: } 1 \text{ unit} = 58.2 \text{ days}$$

$$\text{중력계수: } \mu = 1.0$$

$$\text{초기조건} = [0.0 \quad 1.0 \quad 1.0]^T$$

$$\text{최종점 구속조건} = [0.0 \quad 0.8101 \quad 1.5237]^T$$

최종점 구속조건은 multiplier 를 이용해서 만족시킨다. 그리고 성능함수는 천이하는데 걸리는 시간으로 한다. 그래서 다음과 같이 쓸 수 있다.

$$\text{minimize } J = t_f \quad (43)$$

#### 5. 전산 모의시험 결과

##### 5.1 우주 발사체 문제

파라미터 유도는 2단 초기에 한번만 계산한다고 가정한다. 2단 초기에 3단과 4단에서의 수정량까지 모두 구한다. 그리고 3단과 4단에서는 파라미터의 수정량을 계산하지 않는다. 최종

시간의 고정으로 인해 구속조건 하나가 만족되지 못하는데, 본 논문에서는 속도조건을 완화시켰다.

그림 3 은 최적 제어법칙과 선형 탄젠트 피치 프로그램을 적용했을때 최종 결과를 비교한 것이다. 이때 유상화물의 하중은 각각 864 kg, 858 kg 이다.

그림 4 는 관성 좌표계에서 X 방향의 속도에 변동이 있는 경우에 대해서, 고도조건을 만족시키는 결과를 보여 주고있다. 항력계수에 +10% 오차가 있었을 경우, 2단 연소직전에서 상태 벡터의 오차는

$$e = [ -0.3766 \quad -0.3702 \quad -0.0102 \quad -0.0087 ]^T$$

이다. 이 경우에 단순화된 유도 알고리즘을 적용했을때 피치 프로그램의 시간에 따른 수정량을 그림 5 에서 보여주고 있다. 목표고도에 올라가기 위해 피치를 기준 프로그램보다 더 천천히 증가시킴을 알 수 있다.

## 5.2 궤도 천이 문제

그림 6 은 역시 최적 제어와 파라미터화된 제어 변수의 시간에 따른 값이다. 그림 7 은 초기 반경방향 속도의 변동에 따른 최종점 반경방향 속도를 나타낸다. 초기조건의 변화가 작을 경우에 만족스러운 결과를 보여 주고 있다.

## 6. 결 론

전산 모의시험 결과에서 다음과 같은 결론을 이끌어 낼 수 있다.

첫째, 상태벡터의 변동이 작을 경우에는 implicit 유도는 적절하다. 물론 발사전에 많은 계산이 있어 야지 변동적인 요인을 최대한 줄일 수 있다.

둘째, 단순화된 유도 방식은 적은 양의 데이터 저장을 요구하면서 만족할만한 유도 결과를 보여준다.

파라미터화된 제어 법칙이 비록 suboptimal 이지만 파라미터의 수를 늘림으로서 쉽게 성능을 향상시킬 수 있고 많은 양의 메모리를 필요로 하지 않기 때문에 실제 응용에서 이점이 될 수 있다. 그리고 변동에 대한 유도는 극히 간단하면서도 효과적이다.

## 후 기

본 논문은 과학기술처의 특정연구개발사업 과제 "위성발사체 시뮬레이션 연구(연구책임자: 서울대학교 노 오현 교수)" 의

제 2 세부과제 "유도제어 시스템 연구(책임 연구원: 서울대학교 이 장구 교수)" 의 위탁과제인의 제 2 차년도 연구결과의 일부입니다.

## 참고문헌

- [1] Speyer, J.L. and Bryson Jr., A.E., "A Neighboring Optimum Feedback Control Scheme Based on Estimated Time-to-Go with Application to Re-Entry Flight Paths," *AIAA Journal*, vol.6, 1968, pp. 769-776.
- [2] Bauer, T.P., Wood, L.J., and Caughey, T.K., "Gain Indexing Schemes for Low-Thrust Perturbation Guidance," *Journal of Guidance, Control, and Dynamics*, vol.6, 1983, pp.518-525.
- [3] Chandler, D.C. and Smith, I.E., "Development of the Iterative Guidance Mode with Its Application to Various Vehicles and Missions," *Journal of Spacecraft*, vol.4, 1967, pp.808-903.
- [4] McHenry, R.L., Brand, T.J., Long, A.D., Cockrell, B.F., and Thibodeau III, J.R., "Space Shuttle Ascent Guidance, Navigation, and Control," *The Journal of the Astronautical Sciences*, vol.27, 1979, pp.1-38.
- [5] Sinha, S.K. and Shrivastava, S.K., "Optimal Explicit Guidance of Multistage Launch Vehicle Along Three-Dimensional Trajectory," *Journal of Guidance, Control, and Dynamics*, vol.13, 1990, pp.394-403.
- [6] Bryson Jr., A.E. and Ho, Y.C., *Applied Optimal Control*, Hemisphere Pub., 1975.
- [7] 유창선, "저고도 인공위성의 발사궤적 최적화," KAIST, 1990.
- [8] Gill, P.E., Murray, W. and Wright, M.H., *Practical Optimization*, Academic Press, 1981.
- [9] Williamson, W.E., "Use of Polynomial Approximations to Calculate Suboptimal Controls," *AIAA Journal*, vol.9, 1971, pp.2271-2273.
- [10] Hargraves, C., Johnson, F., Paris, S. and Rettie, I., "Numerical Computation of Optimal Atmospheric Trajectories," *Journal of Guidance, Control, and Dynamics*, vol.4, 1981, pp.406-414.
- [11] 노오현 외, *위성발사체 시뮬레이션 연구 I - 최종연구보고서*, 과학기술처, 1991.
- [12] 탁민제, 진재현, "다단 고체연료 우주발사체의 비행궤적 최적화기법 비교," '91 한국자동제어학회의 논문집(국내편), 1991, pp.413-418.

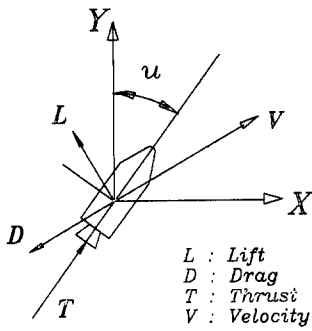


Fig.1 Launcher Dynamic Model

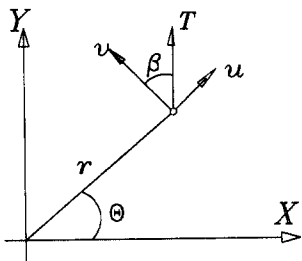


Fig.2 Orbit Transfer Dynamic model

