

논문 2009-46SC-6-11

비선형 하중제어 모델의 예측기 설계 및 알고리즘 구현을 위한 수치연산 오차 분석과 평가

(Realization and Design of Predictor Algorithm and Evaluation of
Numerical Method on Nonlinear Load Control Model)

왕 현 민*, 우 광 준*

(Hyun-Min Wang and Kwang-Joon Woo)

요 약

운동하는 물체를 제어하기 위한 제어이론은 디지털 컴퓨터(임베디드시스템)를 이용하여 복잡한 신경망 이론, 인공지능 이론, 비선형 모델 예측 제어 이론 등이 제어기 설계 단계에서 구현되고 있다. 비행제어 시스템의 비선형 모델 예측 제어 예측기는 구현하는 컴퓨터의 성능과 각종 모듈의 응용프로그램을 하드실시간(Hard Real-Time)으로 처리할 수 있도록 응답 시간을 충족하여야 한다. 이와 동시에 제어 시스템에의 성능을 충분히 발휘할 수 있는 정확성도 고려하여야 한다. 수학적 영역에서의 오류는 전체 알고리즘 구현에 영향을 준다. 그러나 이러한 수학적 오류 발생 요인은 예측기에서 생성되는 파라미터에서 최종 정확도 계산에 가끔 고려하지 않는다. 본 논문에서는 비행체 제어를 위한 디지털 제어 시스템에서 하드실시간 하중제어 모델 예측기를 구현하고, 알고리즘의 응답시간을 살펴본다. 또한 이에 따른 정밀도를 보장하는 고효율 예측기를 구현하는 알고리즘을 살펴본다. 예측기는 하중 제어 모델에서 오일러 방법, Heun 방법, Runge-kutta 방법, 테일러 방법의 수치적분 알고리즘을 사용하여 구현된다.

Abstract

For the shake of control for movement object, control theory like neural network, nonlinear model predictive control(NMPC) is realized on digital high speed computer. Predictor of flight control system(FCS) based nonlinear model predictive control has to be satisfied with response for hard real-time to perform applications on each module in the FCS. Simultaneously, It gives a serious consideration accuracy to give full play to FCS's performance. Error of mathematical aspect affects realization of whole algorithm. But factors of bring mathematical error is not considered to calculate final accuracy on parameter of predictor. In this paper, Predictor was made using load control model on the digital computer for design FCS at hard real-time and is shown response time on realization algorithm. And is shown realization algorithm of high effective predictor over the accuracy. The predictor was realized on the load control model using Euler method, Heun method, Runge-Kutta and Taylor method.

Keywords : Nonlinear Model Predictive Control(NMPC), Predictor, Flight Control System (FCS),
Load Control Model, hard real-time

I. 서 론

운동하는 물체를 제어하기 위한 제어이론은 디지털 컴퓨터(임베디드 시스템)를 이용하여 복잡한 신경망 이

론, 인공지능 이론, 비선형 모델 예측 제어 이론 등이 제어기 설계 단계에서 구현되고 있다.

이러한 디지털 비행 제어 시스템은 알고리즘에 의해 비선형 급변하는 상태에 적응하여 최적의 제어값을 만들어 낼 수 있는 점이 아날로그 시스템과 가장 큰 차이점이라고 하겠다. 이러한 디지털 제어 시스템으로, 최근 모델 기반 예측 제어 시스템의 적용이 다양한 분야에서

* 정회원, 단국대학교 전자공학과
(Dankook University, Dept. of Electronics)
접수일자: 2009년1월23일, 수정완료일: 2009년11월3일

널리 적용되고 있다. 이것은 예측 모델 알고리즘에서 임의의 비선형 함수를 정확하게 표현할 수 있는 성질 때문이다. 즉, 예측기의 성능이 실시간 상태에서 좋은 성능을 발휘한다면, 연속적인 비선형 함수를 원하는 오차 범위 이내로 표현할 수 있다. 이와 같이, 보다 안정화되고 진보된 비행 제어 알고리즘의 실시간 구현에 있어서, 시간에 따라 변화하는 비행 상태 파라미터를 프로그래밍된 알고리즘으로 신속하게 처리할 수 있는 고속의 디지털 컴퓨터가 필요하며, 이를 위한 고효율 수치 해석 알고리즘이 필요하다.

실시간은 실시간 요구 사항을 갖고 있는 응용프로그램을 일컫는 말이다. 이때 실시간 요구 사항들을 엄격히 준수해야하는 응용프로그램을 하드실시간(Hard Real-Time)이라고 한다. 예를 들어 비행체를 제어하는 응용 프로그램들은 지연이 발생할 경우에 엄청난 결과를 초래할 수 있기 때문에 어떠한 이유로든 지연되어서는 안 된다. 또한 비행제어 시스템의 비선형 모델 예측 제어 예측기는 제어기 컴퓨터의 성능과 각종 모듈의 응용프로그램을 하드실시간으로 처리할 수 있도록 응답 시간을 충족하여야 한다. 이와 동시에 제어 시스템에의 성능을 충분히 발휘할 수 있는 정확성도 고려하여야 한다.

비행제어 시스템의 구현에 있어 오차를 야기시키는 요소는 다음과 같이 볼 수 있다. 첫 번째로 움직이는 비행체의 실제 동작에 대한 비행체 형태에 따른, 수학적 모델에서 지정한 파라미터 값의 오차에 의해 발생한다. 이때 제어시스템에서는 설정된 허용오차의 범위 안에 들어야 한다. 두 번째로는 계산 알고리즘에 의한 오차, 즉 파라미터의 보간 방법, 근사화 방법, 수치 적분 방법 및 선형화 모델링 알고리즘에 대한 오차가 야기된다. 세 번째로 알고리즘에 쓰인 수열, 함수 열 및 급수 계산에서의 초기값 설정에 따라 발생하는 오류를 들 수 있다.

수학적 영역에서의 오류는 전체 알고리즘 구현에 영향을 준다. 그러나 이러한 수학적 오류 발생 요인은 예측기에서 생성되는 파라미터에서 최종 정확도 계산에 가끔 고려하지 않는다.

본 논문에서는 비행체 제어를 위한 디지털 제어 시스템에서 하드실시간 하중제어 모델 예측기를 구현하고, 알고리즘의 응답시간을 살펴본다. 또한 이에 따른 정밀도를 보장하는 고효율 예측기를 구현하는 알고리즘을 살펴본다. 예측기는 하중 제어 모델에서 오일러 방법,

Heun 방법, Runge-kutta 방법, 테일러 방법의 수치적분 알고리즘을 사용하여 구현된다. 이렇게 만들어진 예측기 알고리즘은 비선형 모델 예측 제어기(NMPC)에서 예측기 알고리즘으로 실시간으로 구현되어 이용할 수 있다.

II. 비선형 하중 제어 모델에서 예측기 설계

비선형 동역학 시스템을 일반화시킨 식은 다음과 같다.

$$\dot{x} = f(x, u, t, \xi) \quad (1)$$

여기서 x 는 n 계 상태 벡터, u 는 m 계 제어 벡터, $m \leq n$, ξ 는 l 계 외란 벡터, $l \leq n$, $u \in U(t)$, $x \in D_x$ 이고, U 와 D_x 는 각각 m 계, n 계 공간상의 주어진 영역이다. 여기서 초기값 $x_i(0)$, $i = 1, \dots, n$ 는 주어지고, $a \leq t \leq b$ 이다.

본 논문에서 사용하는 하중제어 모델은 다음과 같다.^[1~2]

$$\dot{x}_1 = gn_x - g \sin x_2 \quad (2)$$

$$\dot{x}_2 = \frac{g}{x_1} n_y - \frac{g}{x_1} \cos x_2 \quad (3)$$

$$\dot{x}_3 = -\frac{g}{x_1 \cos x_2} n_z \quad (4)$$

여기서 $x = [V \ \Theta \ \Psi]^T$ 이고, $n = [n_x \ n_y \ n_z]^T$, g 는 중력가속도이다. 이때 모델 식(2),(3),(4)로부터 생성되는 식(1)에서의 f_i ($i = 1, 2, 3$)은 다음과 같이 정해진다.

$$f_1(n_x, x_2) = gn_x - g \sin x_2 \quad (5)$$

$$f_2(n_y, x_1, x_2) = \frac{g}{x_1} n_y - \frac{g}{x_1} \cos x_2 \quad (6)$$

$$f_3(n_z, x_1, x_2) = -\frac{g}{x_1 \cos x_2} n_z \quad (7)$$

이때 위의 비선형 모델식 (2), (3), (4)를 직접, 상태변수 궤환 시스템설계의 관측기는 구현할 수가 없다. 이것은 하중제어 모델이 비선형이고 실시간 변화하는 상태를 포함하고 있기 때문이다. 따라서 이와 같은 비선

형 실시간 예측기 설계는 선형화를 수행하지 않고 수치 해석적 접근이 필요하다.

하중 모델에서 예측기 구현을 위해 비선형인 하중 모델의 미분방정식을 Euler 방법, Heun 방법, Runge-Kutta 방법, Taylor방법을 사용한다. 이러한 미분방정식의 수치해법을 살펴보자.

2.1. Euler Method

비선형 하중 제어 모델에서 Euler 방법은 다음의 예측기 알고리즘은 다음과 같은 식으로 연속적으로 계산된다.

$$x_1(n+1) = x_1(n) + h \cdot f_1(n_x(n), x_2(n)) \quad (8)$$

$$x_2(n+1) = x_2(n) + h \cdot f_2(n_y(n), x_1(n), x_2(n)) \quad (9)$$

$$x_3(n+1) = x_3(n) + h \cdot f_3(n_z, x_1, x_2) \quad (10)$$

2.2. Heun Method

Heun 방법은 Runge-kutta 방법의 반복식이 2차 Taylor 방법의 오차를 갖도록 4개의 파라미터 중 두개는 $\frac{1}{2}$ 로하고 다른 두 개는 1로 했을 때 얻어지는 방법이다.^[3] 이때 하중 제어 모델에서 Heun 방법을 적용하면 $f_i, (i=1,2,3)$ 는 위의 식(4), (5), (6)과 같다.

Heun 방법의 하중제어 모델에서의 예측기 알고리즘은 다음과 같은 식으로 얻어진다.

$$x_1(n+1) = x_1(n) + \frac{h}{2} \left[\begin{matrix} f_1(n_x(n), x_2(n)) \\ + f_1(n_x(n+1), x_2(n)) \end{matrix} \right] \quad (11)$$

$$x_1(n+1) = x_1(n) + \frac{h}{2} \left[\begin{matrix} f_1(n_x(n), x_2(n)) \\ + f_1(n_x(n+1), x_2(n+1)) \end{matrix} \right] \quad (11')$$

$$x_2(n+1) = x_2(n) + \frac{h}{2} \left[\begin{matrix} f_2(n_y(n), x_1(n), x_2(n)) \\ + f_2(n_y(n+1), x_1(n), x_2(n)) \end{matrix} \right] \quad (12)$$

$$x_2(n+1) = x_2(n) + \frac{h}{2} \left[\begin{matrix} f_2(n_y(n), x_1(n), x_2(n)) \\ + f_2(n_y(n+1), x_1(n+1), x_2(n+1)) \end{matrix} \right] \quad (12')$$

$$x_3(n+1) = x_3(n) + \frac{h}{2} \left[\begin{matrix} f_3(n_z(n), x_1(n), x_2(n)) \\ + f_3(n_z(n+1), x_1(n), x_2(n)) \end{matrix} \right] \quad (13)$$

$$x_3(n+1) = x_3(n) + \frac{h}{2} \left[\begin{matrix} f_3(n_z(n), x_1(n), x_2(n)) \\ + f_3(n_z(n+1), x_1(n+1), x_2(n+1)) \end{matrix} \right] \quad (13')$$

위에서 얻어진 Heun 방법의 예측기에서 식(11), (12), (13)에서는 각각의 함수에서 해당하는 함수에 영향이 없는 $x_1(n)$ 또는 $x_2(n)$ 파라미터를 $n+1$ 값이 아닌 n 값을 적용하였다. 반면에 식(11'),(12')(13')에서는 함수에 영향이 직접적으로 없더라도 미세한 변화가 있다고 보고 $n+1$ 의 값을 적용하여 $x_1(n+1)$ 와 $x_2(n+1)$ 로 하였다.

2.3. Runge-Kutta 4차 Method

Runge-Kutta 방법은 반복식에서 증분함수를 적당하게 선택하여 고계 편도함수를 사용하지 않고 향상된 오차를 갖도록 하는 방법이다. Runge-Kutta 4차는 다음의 식에서 알 수 있듯이 Heun방법 보다 계산량이 증가한다. Runge-Kutta 4차 방법의 하중제어 모델의 예측기 알고리즘은 다음과 같은 식으로 얻어진다. 먼저 하중 제어 모델 $f_i, (i=1,2,3)$ 에서 첫 번째 식은 다음과 같다.

$$x_1(n+1) = x_1(n) + \frac{1}{6} [F_1 + 2F_2 + 2F_3 + F_4] \quad (14)$$

$$F_1 = hf(n_x(n), x_2(n)), \quad F_2 = hf\left(n_x(n) + \frac{1}{2}h, x_2(n)\right)$$

$$F_3 = hf\left(n_x(n) + \frac{1}{2}h, x_2(n)\right), \quad F_4 = hf(n_x(n) + h, x_2(n))$$

식(14)에서 $x_2(n)$ 이 $x_1(n)$ 의 원소이나 $x_2(n)$ 의 도함수값 0이므로 $F_i, (i=1,2,3,4)$ 에서 $x_2(n)$ 에 이전의 증감을 더해주지 않는다. 두 번째로 하중 제어 모델 f_2 에서 Runge-Kutta 4차 식은 다음과 같다.

$$x_2(n+1) = x_2(n) + \frac{1}{6} [F_1 + 2F_2 + 2F_3 + F_4] \quad (15)$$

$$F_1 = hf(n_y(n), x_1(n), x_2(n))$$

$$F_2 = hf\left(n_y(n) + \frac{1}{2}h, x_1(n), x_2(n) + \frac{1}{2}F_1\right)$$

$$F_3 = hf\left(n_y(n) + \frac{1}{2}h, x_1(n), x_2(n) + \frac{1}{2}F_2\right)$$

$$F_4 = hf(n_y(n) + h, x_1(n), x_2(n) + F_3)$$

식(15)에서 $x_2(n)$ 이 $x_2(n)$ 의 원소이고 $x_2(n)$ 의 도함수값 0이 아니므로 $F_i(i=1,2,3,4)$ 에서 $x_2(n)$ 에 이전의 증감을 더해준다. 하중 제어 모델 세 번째 식 f_3 에서 식은 다음과 같다.

$$x_3(n+1) = x_3(n) + \frac{1}{6} [F_1 + 2F_2 + 2F_3 + F_4] \quad (16)$$

$$F_1 = hf(n_z(n), x_1(n), x_2(n))$$

$$F_2 = hf\left(n_z(n) + \frac{1}{2}h, x_1(n), x_2(n)\right)$$

$$F_3 = hf\left(n_z(n) + \frac{1}{2}h, x_1(n), x_2(n)\right)$$

$$F_4 = hf(n_z(n) + h, x_1(n), x_2(n))$$

2.4. Taylor 2차 Method

Taylor 방법은 차수가 높으면 오차가 작아지나 고계 편도함수를 구해야 하는 단점이 있다. 이에 따라 계산 시간이 증가한다.

$$y_{n+1} = y_n + hT_k(t_n, y_n; h) \quad (17)$$

$$T_k(t, y; h) = f(t, y) + \dots + \frac{h^{k-1}}{k!} f^{(k-1)}(t, y)$$

$$f'(t, y) = f_t(t, y) + f_y(t, y)f(t, y)$$

하중제어 모델에서 Taylor방법을 적용하기 위한 편도함수는 다음과 같다.

$$f'_1(n_x, x_2) = f_{n_x}(n_x, x_2) + f_{x_2}(n_x, x_2)x'_2 \quad (18)$$

$$f'_2(n_y, x_1, x_2) = f_{n_y}(n_y, x_1, x_2) + f_{x_1}(n_y, x_1, x_2)x'_1 + f_{x_2}(n_y, x_1, x_2)x'_2 \quad (19)$$

$$f'_3(n_z, x_1, x_2) = f_{n_z}(n_z, x_1, x_2) + f_{x_1}(n_z, x_1, x_2)x'_1 + f_{x_2}(n_z, x_1, x_2)x'_2 \quad (20)$$

따라서 Taylor방법에서의 구현 알고리즘은 다음과 같다.

$$x_1(n+1) = x_1(n) + h \left[f(n_x(n), x_2(n)) + \frac{h}{2} f'(n_x(n), x_2(n)) \right] \quad (21)$$

$$x_2(n+1) = x_2(n) + h \left[f(n_y(n), x_1(n), n_z(n)) + \frac{h}{2} f'(n_y(n), x_1(n), x_2(n)) \right] \quad (22)$$

$$x_3(n+1) = n_3(n) + h \left[f(n_z(n), x_1(n), x_2(n)) + f'(n_z(n), x_1(n), x_2(n)) \right] \quad (23)$$

III. 알고리즘 실험 및 특성 분석

3.1. 알고리즘 성능 분석

표 1은 각각의 수치 해법을 구현하여 계산 시간과 예측 시간을 나타내었다. 알고리즘 구현은 펜티엄 2.2GHz, RAM 1G Byte에서 매트랩 프로그램으로 수행되었다. 표에서 N 은 분해구간 수이며, $\Delta t_{cputime}^{Matlab}$ 은 매트랩에서 프로그램을 한번 수행하여 응답을 구하는데 걸리는 실제 시간이다. 또한 프로그램 수행에서 단위 시간당 예측 T_p 시간이다.

예측기 성능을 판별하기 위해 다음의 식(24)와 같이 예측기 성능 지수를 정의 한다. 이러한 예측기 성능 지수는 JM(Joon & Min) 성능지수라 부른다.

$$I_p = \Delta \times T_p \quad (24)$$

여기서 Δ 는 수치해석 알고리즘에서 한 스텝을 처리

표 1. 계산 시간과 예측 시간

Table 1. Calculation Time and Prediction Time.

Method	$\Delta t_{cputime}^{Matlab}$ [sec]	N	[sec/ N]	T_p
Euler	3.36784	360	0.0094	106.8934391
Heun	6.10577	360	0.0169	58.96062249
Runge-Kutta (4th)	7.45961	360	0.0207	48.25989562
Taylor (2nd)	5.21750	360	0.0145	68.99856253

표 2. JM 성능지수

Table 2 JM Performance Index.

Method	[sec/ N]	T_p	I_p
Euler	0.0094	106.8934391	1.004798328
Heun	0.0169	58.96062249	0.99643452
Runge-Kutta (4th)	0.0207	48.25989562	0.998979839
Taylor(2nd)	0.0145	68.99856253	1.000479157

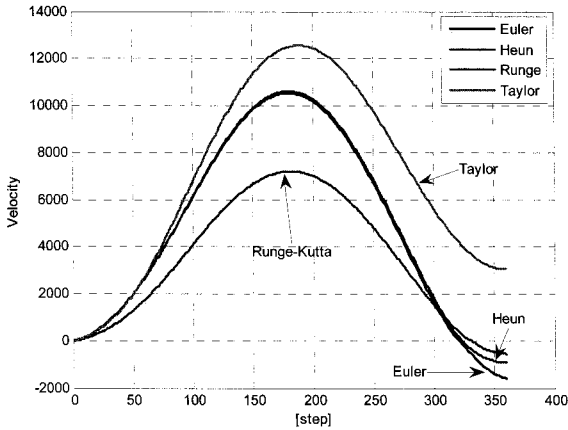


그림 1. 하중제어 모델의 예측기 속도 변화
Fig. 1. Change Velocity of Predictor on Load Control Mode.

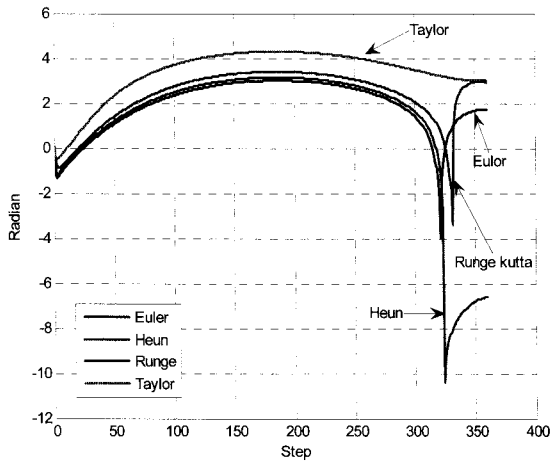


그림 2. θ 값의 변화
Fig. 2. Change θ of Predictor on Load Control Model.

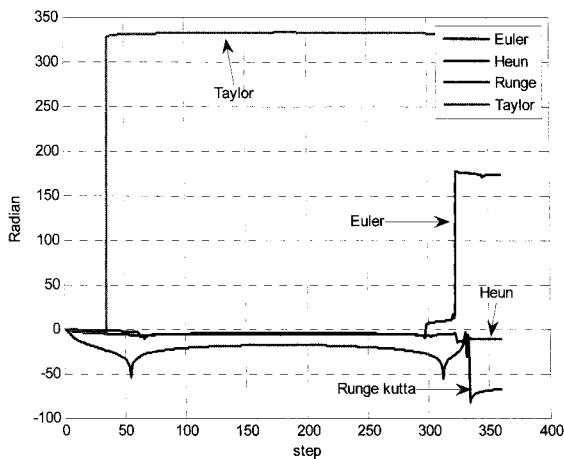


그림 3. ψ 값의 변화
Fig. 3. Change ψ of Predictor on Load Control Model.

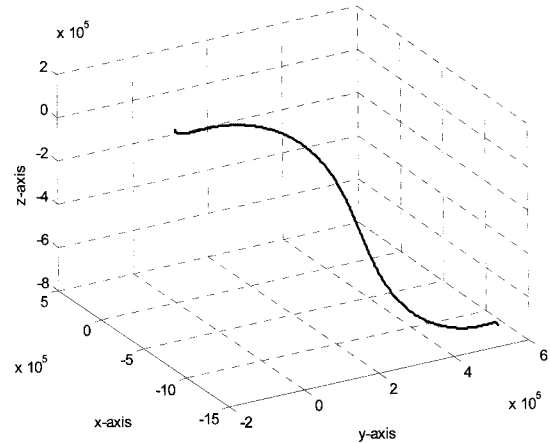


그림 4. 하중제어 모델의 Heun방법에 의해 구현된 경로 $n_x, n_y, n_z = \sin$ 함수
Fig. 4. Trajectory of Heun's Method on Load Control Model at $n_x, n_y, n_z = \sin$ Function.

하는데 걸리는 시간이고 T_p 는 단위 시간당 예측시간이다. JM성능 지수는 실시간 예측기 제작에서 작은 값일 수록 좋은 성능으로 판별한다. 표 2에서 알 수 있듯이 JM성능지수가 가장 좋은 것은 Heun 방법이다.

다음의 그림 1, 2, 3에서는 각각의 방법에서 1000[m/s]로 던져지는 물체에 각각의 축에 sin의 하중이 입력될 때의 V, θ, ψ 값을 각각 나타내었다.

그림 1에서 Heun 방법은 Euler 방법과 유사한 형태이고 Taylor 2차, Runge-Kutta 4차와는 많은 차이가 있다. 또한 그림 2에서는 속도가 음(-)으로 변화에 따른 급격한 반응각 변화를 알수가 있다. 그림 3은 경로각을 나타내는데 Heun방법을 제외하고는 현저한 급격한 증가를 보였다.

그림 4는 초기 좌표가 (1000, 1000, 1000)에서 초기 속도 10000[m/s]로 던져진다. 이때 운동하는 동안 n_x, n_y, n_z 에 sin 형태의 하중이 작용할 때 경로를 나타낸 것이다. 그림4는 Heun 방법에 의해 구현된 경로이다.

3.2. 알고리즘의 하중 제어 모델에서의 자유 낙하 예측기 구현

뉴우턴 법칙에 의한 자유롭게 낙하하는 물체에 대한 지상에 도달하는 거리 $R(T)$ 와 시간 (T) 은 다음과 같다.

$$R(T) = \frac{V_0^2}{g} \sin(2\alpha) \tag{25}$$

$$T = \frac{2V_0}{g} \sin \alpha \quad (26)$$

여기서 V_0 은 던져지는 속도[m/s], α 는 던져지는 지상과 이루는 각도[°], g 는 중력가속도이다.

식(25), (26)에 의해, 속도 V_0 가 10000[m/s]이고 $\alpha = 1^\circ$ 일때 도달거리는 356117.0[m]이고 시간은 35.6171[sec]이다.

다음으로 Heun 방법에서의 예측기를 구현해 보자. 초기 던져지는 속도 $V_0 = 10000$ [m/s]이고 던져질때 각도는 $\alpha = 1^\circ$, 초기 던져지는 공간상의 좌표는 (0, 0, 0)이다.

Heun 방법에서 시물레이션은 N 값이 10, 100, 1000, 40, 400일 때 구현해 보았다. 각각의 운동 경로를 그림5에 나타내었다. 그림에서는 $N = 1000$ 일때 거리와 시간의 오차가 가장 작았다.

표 3에서 Heun 알고리즘 구현에서 N 에 따른 JM성

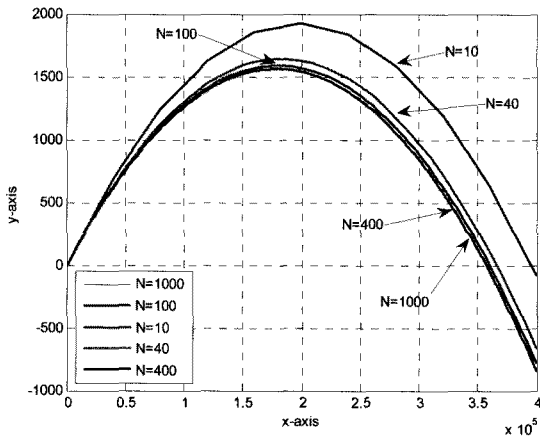


그림 5. N값에 따른 Heun 알고리즘에서의 경로
Fig. 5. Trajectory of Heun Algorithm according to N.

표 3. Heun방법의 예측기 알고리즘에서 성능 지수
Table 3. Performance Index of Heun's Predictor Algorithm.

N	T_p [sec]	거리 [m]	계산 시간 [sec]	오차		J_p^{Heun}
				거리	시간	
1000	1.82039	356520	35.64	-402.98	-0.0229	0.0356
100	20.4251	360110	36	-3992.99	-0.3829	0.36
10	173.663	400000	40	-43883.0	-4.3829	4
40	47.9309	366050	36	-9932.99	-0.3829	0.9
400	5.14410	357120	35.7	-1002.99	-0.0829	0.0892

능지수를 나타내었다. JM 성능 지수가 가장 작을 때 오차가 최소임을 알 수 있다. 본 자유 낙하 실험을 위해 하중제어 모델에서 예측기의 Heun 알고리즘 구현은 표 3과 같이 JM성능지수가 향상될수록 오차가 감소함을 알 수 있다.

IV. 결 론

비행체 디지털 제어 시스템 제작을 위해, 다중화 처리를 실시간 구현은 시스템의 하드웨어(H/W) 성능에 맞는 최적의 수행 속도와 오차를 구현하는 알고리즘을 선택하여야 한다.

본 논문에서는 비선형 하중제어 모델의 예측기 구현을 위해 수치해석 방법을 사용하여 실시간 알고리즘 만들었다. 각각의 알고리즘에 따라 성능지수를 판별하였으면, 구현되는 알고리즘에서 N값에 따른 성능지수를 정의하였다.

본 논문에서 제시한 JM성능지수에 근거한 수치해석 방법의 선택과 하중제어 모델에서의 알고리즘 구현은 실시간 비행 제어 시스템의 예측기 알고리즘 설계를 보다 효율적으로 하는데 기여 할 것이다. 또한 이러한 방법은 비선형 모델 예측 제어 시스템에서 유용하게 사용할 수 있을 것이다. 또한 슈퍼 컴퓨터에서 비행 경로 예측 통합 시스템 구축에 유용한 기초가 될 것이다.

참 고 문 헌

- [1] 왕현민, 우광준, 허경무, "비행 물체의 유도제어 시스템 설계를 위한 하중(중력수) 제어 모델의 성능 분석", 전자공학회 논문지, 제46권, 제1호, 2009. 1.
- [2] 왕현민, "움직이는 물체의 운동 하중제어 모델과 경로 특성(M-설계방법)", 제어·자동화·시스템 공학회 대전충청지부 학술대회 논문집, 한국, pp.62-66, 2007. 12.
- [3] 이규봉, "수치해석학", 경문사, 2007.
- [4] I. A. Maron, N. V. Kopchenova, "Computational Mathematics", Mir Publishers, 1989.
- [5] W.F. Hornyak, J.B. Marison, "Physics for Science and Engineering", Maryland university, 1982.
- [6] 송주한, 안대순, 오상진, 박광규, "자유진동 해석에서 수치적분기법의 수치적평가", 한국소음진동학회 2001년, 추계학술대회논문집, pp.1249-1253, 2001.
- [7] N. Baxvalov, N. Zhidkov, G. Kobalkov, "Numerical Method", Fizimalit, Saint-Peterburg, Russia, 2001.

- [8] 탁민제, 방효충, 김유단, 홍성경, 김병수, “비행동역학 및 제어”, 경문사, 1판, 2004.
- [9] J.M. Maciejowski, “Predictive Control with Constrains”, Prentice Hall, 2002.
- [10] K.K. Jang, Y.H. Joo, O.S. Kwon, “Intelligent Digital Redesign of a Fuzzy-Model-Based Controllers for Nonlinear System with Uncertainties”, JCASE, vol. 12, No. 3, pp.227-232, March, 2006.
- [11] Richard C. Dorf, Robert H. Bishop, Modern Control Systems, 11th ed., Prentice Hall, 2008.

저 자 소 개



왕 현 민(정회원)-교신저자
 1996년 단국대학교 전자공학과 졸업.
 2000년 Saint-Petersburg state of aerospace instrumentation, 제어시스템 석사 졸업.
 2005년 Baltic State Technical university, 제어시스템 박사 졸업.

우 광 준(정회원)
 대한전자공학회 논문지 제 44 권 SC편 2007년 11월 제 6 호 참조

<주관심분야 : 최적제어, 우주기술응용, 비선형 제어>