

## 技術論文

J. of The Korean Society for Aeronautical and Space Sciences 42(10), 884-892(2014)

DOI: <http://dx.doi.org/10.5139/JKSAS.2014.42.10.884>

## 메타모델(Meta Model)을 활용한 시뮬레이터 구현충실도 향상 연구

조동현\*, 권기범\*, 실현주\*, 명현삼\*\*, 장영찬\*\*

A Study on Performance Enhancement in Simulation Fidelity  
Using a Meta ModelDonghyurn Cho\*, Kybeom Kwon\*, Hyunju Seol\*, Hyunsam Myung\*\* and YoungChan Chang\*\*  
Republic of Korea Air Force Academy\*, Agency for Defence Development\*\*

## ABSTRACT

In this paper, a meta model using neural network substituting for the simulator aerodynamic database is proposed to improve simulation fidelity near the critical flight area and real-time performance. It is shown that the accuracy of the meta model is relatively higher than the existing table lookup methods for arbitrary nonlinear database and the calculation speed is also improved for a specific F-16 maneuver simulation. The increase in the number of hidden nodes in the meta model for better accuracy of database representations causes a delay in function generation due to increased time required for computing exponential functions. In order to make up this drawback, we additionally study the fast exponential function method.

## 초 록

비행 시뮬레이터에 사용되는 데이터베이스를 신경회로망을 이용한 메타모델로 대체함으로써 기존 시뮬레이터 대비 비행 한계점 부근 공력특성에 대한 구현충실도 및 실시간성을 향상시키고자 하였다. 임의의 비선형 데이터베이스에 대해 메타모델의 정확도가 기존 테이블 검색 방식에 비해 상대적으로 높음을 확인하였으며 특정 F-16 기동에 대한 시뮬레이션을 통해 계산속도도 향상됨을 확인하였다. 신경회로망을 이용한 메타모델에서 은닉노드 수의 증가는 데이터베이스에 대한 추정정확도를 향상시키지만 많은 계산 시간이 소요되는 지수함수 수의 증가로 인해 함수발생 속도가 지연되게 된다. 이를 보완하기 위해 빠른 지수함수 계산 방법을 적용하는 연구도 추가적으로 수행하였다.

**Key Words** : Function Generation(함수발생), Meta Model(메타모델), Neural Network(신경회로망), Real-Time Simulation(실시간 시뮬레이션), Table Lookup(테이블 검색)

## 1. 서 론

비행시뮬레이터(Flight Simulator)의 구현충실도(Fidelity)는 비행성능, 조종성, 항공기의 시스

템, 계기반응 및 제어반응 등 기본적인 비행모의와 함께, 계통 및 운용환경 모의와 같은 광범위한 영역에서 수행되는 비행시뮬레이터 설계 및 제작 목표라 할 수 있다. 많은 기술향상에도 불

† Received: December 3, 2013 Accepted: September 22, 2014

\* Corresponding author, E-mail : kybeom.kwon@gmail.com

<http://journal.ksas.or.kr/>

pISSN 1225-1348 / eISSN 2287-6871

구하고 실속(Stall), 이착륙과 같은 제한 한계점에서 낮은 구현충실도는 여전히 시뮬레이터의 큰 단점 중 하나이다[1].

풍동실험 및 비행시험을 통해 획득되는 공력 데이터베이스는 구현충실도에 가장 큰 영향을 주는 요소로서 일반적으로 테이블의 형태로 주어진다. 주어진 비행조건에서 공력계수를 추출하는 것을 함수발생(Function Generation)이라고 하며, 상용 시뮬레이터의 함수발생 기법은 테이블 검색(Table Lookup)을 통해 받음각, 옆미끄럼각, 마하수, 고도 등의 비행조건에 해당하는 독립변수들의 구간을 찾고, 공력계수에 대한 선형보간(Linear Interpolation)을 수행하게 된다. 비행 한계점 부근의 공력특성은 일반적으로 비선형이므로 선형보간을 통한 방법은 구현충실도를 저하시킬 위험성이 존재한다.

또한, 최근 가상 군 훈련체계 분야에서 각광을 받고 있는 LVC(Live-Virtual-Constructive) 환경은 각 시뮬레이션의 장단점을 보완하고, 시스템적 관점에서의 훈련을 통해 저비용, 고효율의 교육훈련이 가능한 장점은 있으나[2] 단일 시뮬레이터가 수행하는 기본작업 외에도 각 시스템 간 통신, 각 개체의 동기화를 위한 해상도 변환, 다수의 Virtual 객체 간 동시 시뮬레이션 등의 부가적인 작업이 단위 시간간격동안 수행되어야 하므로 실시간 시뮬레이션을 위해서는 일련의 작업들이 훨씬 신속하게 수행되어야 한다.

본 논문에서는 주어진 데이터베이스를 기본적으로 비선형 모델로 생성되는 메타모델(Meta Model)로 대체할 경우 비행 한계점 부근의 공력 비선형성을 실제와 가깝게 구현할 수 있는 가능성을 제시하고자 하였다. 또한, 단일 시뮬레이터 및 LVC 환경에서 공력특성에 대한 구현충실도를 유지하면서 실시간 시뮬레이션을 보장하고, 부가적으로 영상구현과 같은 다른 시뮬레이션 큐(Queue)의 구현충실도 향상을 위한 여유를 제공할 수 있도록 메타모델의 함수발생 소요시간을 감소시키기 위한 연구를 수행하였다.

메타모델을 생성하기 위한 방법은 여러 가지가 있으나 본 논문에서는 신경회로망(Neural Network) 기법을 적용하였다. 신경회로망 기법은 인간 두뇌구조의 작동원리를 모방한 지도학습 기법으로 입력 값과 출력 값들이 비선형인 경우 데이터를 추정하는데 효과적인 방법이다[3]. 신경회로망 방법의 적용 시 상대적으로 많은 시간이 소요되는 지수함수(Exponential Function)의 계산 시간 단축을 위해 컴퓨터에서 흔히 적용되는 IEEE-754에서 규정된 실수 표현의 메모리 구조를 이용하는 방법도 시도하였다[4].

## II. 본 론

### 2.1 함수발생 기법

#### 2.1.1 테이블 검색(Table Lookup) 및 보간

일반적으로 비행 시뮬레이터에서 사용되는 함수발생 기법은 주어진 데이터베이스에서 해당 조건을 만족시키는 데이터 구간을 검색하는 과정과 검색된 단위 데이터 구간 끝단의 값들로부터 보간을 통해 해당 데이터를 획득하는 두 부분으로 구분되어진다. 후자의 경우 검색된 단위 구간 끝단의 데이터 값들에 대한 선형적 관계를 이용해 계산되어진다.

상용 시뮬레이터에서 흔히 이용되는 단위구간을 검색하는 방법으로는 데이터의 독립변수 초기 값부터 순차적(Sequential)으로 검색하지만 검색 시간을 단축하기 위해 이분검색법, 보간검색법 및 검색창 검색법(Dynamic Window Search) 등이 기존의 연구에서 제시되었다[5-6]. 이분검색법은 찾고자 하는 단위구간을 정하기 위해서 전체 데이터 테이블에서 원하는 단위구간까지 데이터 검색의 크기를 절반씩 줄여가며 해당하는 단위구간을 찾는 방법이다. 보간검색법은 일차방정식의 해를 구하는 것과 동일한 방법으로 데이터 값들이 선형적으로 분포되어 있다는 가정 하에 데이터 테이블 양 끝단의 데이터 값들을 직선으로 연결한 선형라인 검색을 통해 단위구간을 결정하는 방법이다. 이분검색법이 데이터 형태와 무관하게 데이터 수에 의해서 검색속도가 결정되는 반면 보간검색법은 데이터 형상에 따라 검색속도가 결정되어지는 경향이 있다. 예로서 데이터가 선형적으로 분포되어 있을수록 보간검색법을 통해 빠른 검색속도로 단위구간을 결정할 수 있지만, 데이터 형상이 비선형적으로 분포되어 있다면 보간검색법의 데이터 검색속도는 현저하게 저하되어진다. 그러므로 검색방법을 선정할 때 데이터의 수뿐만 아니라 데이터의 형상도 포함하여 고려해야 한다. 마지막으로 검색창 검색법은 시스템의 특성을 이용하여 데이터 테이블의 검색영역을 축소시킴으로써 검색속도를 향상시키는 방법이다. 일반적으로 다음 시간단계에서의 함수발생 시 의도된 급격한 기동이나 외란이 가해지지 않는 한 이전의 단위구간에서 크게 변화되어지지 않기 때문에 이전의 단위구간 부근으로 데이터 검색영역을 축소시킬 수 있다. 이 검색법은 검색할 데이터양이 방대하다 하더라도 일정한 패턴 또는 특성을 가지고 데이터가 변할 때 이러한 특징을 이

용하여 검색 영역을 축소시킴으로써 검색속도를 향상시킬 수 있다.

테이블 검색기법을 활용한 다양한 함수발생기법이 연구되었으나 최종 함수발생을 위해서는 모두 검색된 단위구간에서 선형보간을 이용한다.

2.1.2 메타모델(Meta Model)

메타모델을 생성할 수 있는 방법 중 신경회로망 기법은 인간 두뇌구조의 작동원리를 모방한 지도학습 기법으로 한 개의 뉴런(Neuron)들을 상호 연결하여 입력값에 대한 최적의 출력값을 예측하는 것이다. 즉, 입력값과 출력값 사이의 비선형 관계를 모형화하는 방법이다. Fig. 1은 전형적인 신경회로망을 간단하게 표현한 것이다. 입력층(Input Layer), 은닉층(Hidden Layer), 출력층(Output Layer)의 3개층으로 구성되어 있고, 각 층마다 다수의 뉴런을 포함하고 있다. Fig. 1에서 원으로 표시된 것은 은닉노드(Hidden Node)라고 하는데 입력노드가 자극(입력치)을 접수하면 은닉노드가 이를 전달받아 중간에서 처리하고 최종적으로 출력노드가 반응하게 된다. 즉, 입력층에서 독립변수의 값을 받고 출력층에서 종속변수의 값을 출력하게 된다.

메타모델에 의한 함수발생은 데이터베이스를 대체한 신경회로망 모델을 통해 주어진 비행조건을 독립변수로 받아 검색작업 없이 공력계수를 바로 출력하게 된다.

2.2 메타모델 생성

실속 이후 및 이착륙과 같은 비행한계점 부근에서 공력특성의 비선형성을 유지하는 것은 구현 충실도 향상에 중요하다. Fig. 2에서 일반적으로 사용하는 테이블 검색 방식을 통해 추정된 데이터와 실제 데이터를 도시하였다.

Figure 2에서 보듯이 선형구간 내에서 테이블 검색 방식을 통한 데이터 추정은 신뢰할 수 있지만, 비선형성이 높은 구간에서 테이블 검색 방식을 사용한 데이터 추정은 상당한 크기의 오차를 발생시킬 수 있다.

그러므로 비선형성을 보유한 메타모델을 통해

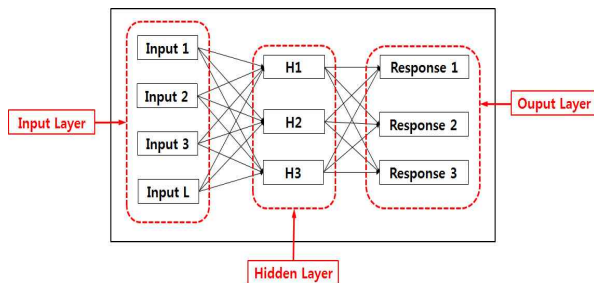


Fig. 1. Schematic of Neural Network

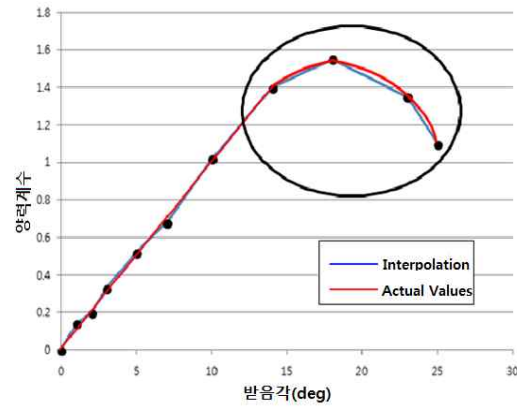


Fig. 2. The Error in Data Estimation by using Table Lookup Method

데이터테이블을 대체함으로써 비선형성이 높은 구간에서도 신뢰성 있게 데이터를 추정할 수 있다.

2.2.1 신경회로망 기법을 통한 메타모델 생성

신경회로망 기법은 고차의 비선형 문제에 대한 데이터 추정에 적합한 방법이며 이산적인 문제에도 쉽게 적용할 수 있다. 신경회로망에서 기본 모델로 활용되는 Logistic Sigmoid 함수는 식(1)과 같이 표시할 수 있고, 이 함수에 입력값,  $z$ 를 받으면 출력값은 Squishing되어 항상 0과 1 사이의 값으로 나타난다.

$$S(z) = \frac{1}{1 + e^{-z}} \tag{1}$$

신경회로망의 최종형태는 식(2)와 같이 표현되어진다.

$$R_k = e_k + \sum_{j=1}^{N_H} \left[ f_{jk} \left( \frac{1}{1 + e^{-\left( a_j + \sum_{i=1}^N b_{ij} X_i \right)}} \right) \right] \tag{2}$$

여기서,

- $a_j$  : the intercept term for the  $j^{\text{th}}$  hidden node
- $b_{ij}$  : the coefficient for the  $i^{\text{th}}$  design variable
- $X_i$  : the value of the  $i^{\text{th}}$  design variable
- $N$  : the number of input variables
- $e_k$  : the intercept term for the  $k^{\text{th}}$  response
- $f_{jk}$  : the coefficient for the  $j^{\text{th}}$  hidden node and  $k^{\text{th}}$  response
- $N_H$  : the number of hidden nodes
- $R_k$  : the value of the  $k^{\text{th}}$  response

식(2)의 계수들인  $a_i, b_{ij}, e_k, f_{jk}$  는 주어진

공력데이터 테이블인 Training 데이터로부터 종속변수 값을 추정할 수 있는 최적화를 수행함으로써 결정되어진다. 최적화 지수는 주로 RMS(Root Mean Square) 오차가 되며, 다양한 Training 알고리즘을 적용할 수 있다.

### 2.3 데이터추정 정확도 비교

데이터 추정정확도를 비교하기 위해서 임의로 선정된 데이터베이스를 선형형태에서 비선형 형태로 점진적으로 변화시키면서, 테이블 검색 방식과 신경회로망 기법을 적용한 메타모델을 통해 얻어진 데이터추정 값과 실제 값들의 오차를 비교하여 데이터 추정정확도를 분석하였다. 분석을 위해 선정된 데이터베이스는 받음각과 마하수에 대한 양력계수( $C_L$ ) 함수로 아래 3가지 형태로 데이터 함수를 가정하였다. 다양한 데이터 형태에서 데이터 추정정확도를 비교하기 위해서 선형함수 형태에서 주기함수 및 지수함수를 추가하면서 비선형성이 부가됨에 따른 데이터 추정정확도를 확인하였다.

Case 1) 선형함수 형태의 데이터함수

$$C_L(\alpha(i), M(j)) = C_a\alpha(i) + C_mM(j)$$

Case 2) 주기함수 형태의 데이터함수

$$C_L(\alpha(i), M(j)) = C_a\alpha(i) + C_mM(j) + C_{ac}\cos(\alpha(i)) + C_{ms}\sin(M(j))$$

Case 3) exp함수 형태의 데이터함수

$$C_L(\alpha(i), M(j)) = C_a\alpha(i) + C_mM(j) + C_{ac}\cos(\alpha(i)) + C_{ms}\sin(M(j)) + C_{ae}\exp(\alpha(i)) + C_{me}\exp(M(j))$$

각 경우의 데이터 함수에 대한 데이터 테이블을 만들기 위해 받음각( $\alpha$ )은 최소  $0^\circ$ 에서  $2^\circ$ 간격으로 최대  $20^\circ$ 까지 11개의 데이터들로 구성하였고, 마하수( $M$ )는 최소 0에서 0.2 간격으로 최대 1까지 6개의 데이터로 총 데이터 테이블은 받음각( $\alpha$ )과 마하수( $M$ )의 변수로 구성된 크기  $11 \times 6$  행렬의 66개의 데이터들로 구성되어진다. 데이터 함수에 사용된 계수는 Table 1과 같다.

Table 1의 계수값을 이용하여 데이터 테이블을 구성한 후에 테이블 검색 방식과 신경회로망

Table 1. Coefficients used in Data Function

계수	$C_a$	$C_m$	$C_{ac}$	$C_{ms}$	$C_{ae}$	$C_{me}$
값	1	1	0.2	0.2	0.1	0.1

을 적용한 메타모델을 통해 각각 임의의 100개의 받음각과 마하수에 대한 양력계수 값을 추정하였다.

이렇게 추정된 데이터 값과 주어진 데이터 함수의 실제값 사이의 오차를 Fig. 3-4에 도시하였다. Fig. 3은 테이블 검색 방식에 대한 데이터 추정오차를 보여준다. 테이블 검색 방식을 통한 단위 구간 검색 기법은 이분검색법, 보간검색법, 검색창 검색법을 사용하였으나 이 방법들은 단위 데이터 구간을 검색하는 과정에서만 차이가 있을 뿐, 일단 단위 데이터 구간이 결정되어지면 단위 데이터 구간사이의 데이터 값들을 통해 선형보간법을 이용하여 데이터를 추정하기 때문에 각각의 테이블 검색 방식을 활용한 기법들은 동일한 데이터 추정값을 가진다. Fig. 4는 신경회로망을 적용한 메타모델을 통한 데이터추정 오차 결과를 보여주고 있다. 데이터 추정에 적용한 신경회로

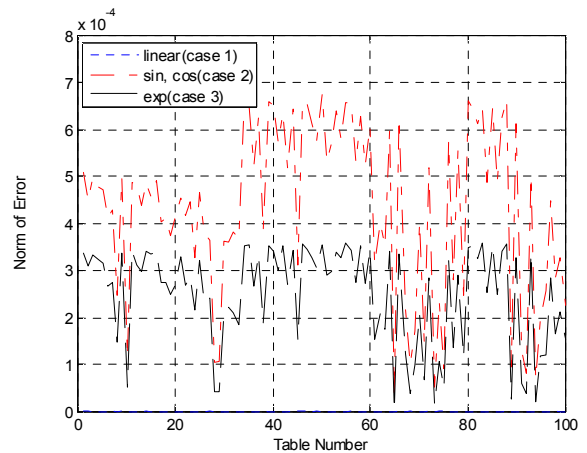


Fig. 3. The Error in Data Estimation (Table Lookup)

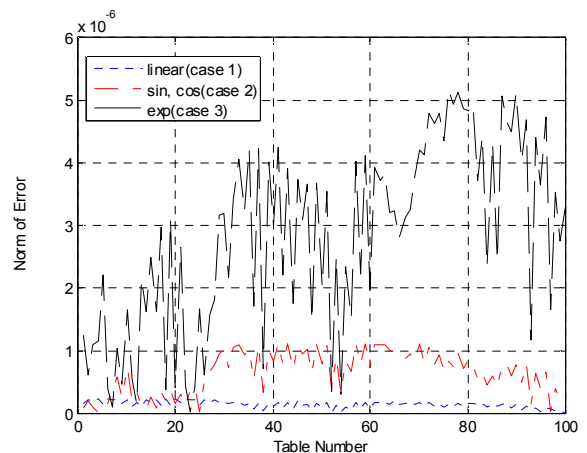


Fig. 4. The Error in Data Estimation (Meta Model)

망의 은닉노드 수는 5개이다.

Figure 3과 4를 보면 선형함수 형태의 데이터 (Case 1)에서 테이블 검색 방식을 통한 데이터 추정은 실제값과 거의 동일한 값을 추정하지만 Case 2), Case 3)과 같은 비선형 형태의 데이터 테이블에 대해서는  $10^{-4}$  정도의 오차가 발생하는 것을 확인할 수 있다. 하지만 신경회로망 메타모델을 이용한 데이터 추정은 데이터의 형태와 무관하게  $10^{-6}$  정도로 테이블 검색 방식과 비교해 볼 때 약 100배정도 작은 수준의 오차가 발생하는 것을 확인할 수 있다.

2.4 함수발생 속도 비교

테이블 검색방법에 의한 함수발생은 주어진 데이터 수와 형태뿐만 아니라 어떠한 검색기법을 선택하느냐에 따라 함수발생 속도가 달라지며 검색시간을 단축하기 위해 기존에 제안된 검색창 검색법은 고성능 전투기 시뮬레이터에서 급기동 시 주어진 검색창 반경을 벗어날 위험성도 상존한다. 신경회로망 메타모델을 이용할 경우 데이터 수 및 형태와 관계없이 은닉노드의 수만 같으면 동일한 함수발생시간을 나타내며 항상 데이터 베이스 내의 함수값을 출력한다.

함수발생 속도를 비교하기 위해 F-16 전투기의 무게중심이 0.35 평균코드 위치의 트림상태에서 마하수와 받음각에 따른 양력계수 데이터베이스를 사용하였다[7]. 양력계수는 받음각  $0^\circ, 1^\circ, 2^\circ, 3^\circ, 5^\circ, 7^\circ, 9^\circ, 13^\circ, 17^\circ, 21^\circ$ 의 10개의 키 값과 마하수의 경우 0.5, 0.6, 0.7, 0.8, 0.85, 0.9의 6개의 키 값으로 총  $10 \times 6$  크기의 60개의 데이터 테이블을 구성하였다. 주어진 데이터 테이블은 Table 2에 나타내었다.

Table 2. The Lift Coefficients( $C_L$ ) of F-16 Aircraft for  $\alpha$  and  $M$

$M \backslash \alpha$	0.5	0.6	0.7	0.8	0.85	0.9
$0^\circ$	0.025	0.019	0.012	0.004	-0.009	-0.021
$1^\circ$	0.089	0.085	0.085	0.085	0.079	0.076
$2^\circ$	0.155	0.155	0.157	0.161	0.167	0.173
$3^\circ$	0.222	0.224	0.229	0.239	0.255	0.274
$5^\circ$	0.367	0.365	0.371	0.394	0.420	0.442
$7^\circ$	0.507	0.509	0.510	0.544	0.557	0.566
$9^\circ$	0.623	0.626	0.633	0.656	0.665	0.668
$13^\circ$	0.849	0.837	0.796	0.822	0.843	0.846
$17^\circ$	1.091	1.070	1.024	1.027	1.017	0.999
$21^\circ$	1.281	1.264	1.240	1.224	1.206	1.189

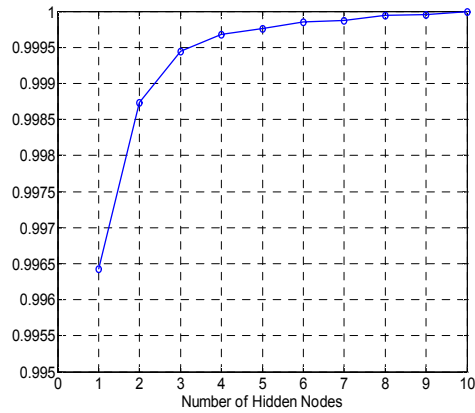


Fig. 5.  $R^2$  Variation by Number of Hidden Nodes

2.4.1 신경회로망 모델

데이터 추정을 위한 메타모델 생성을 위해 신경회로망 모델의 은닉노드를 1~10개로 순차적으로 변경하며 Training을 수행하였다. 일반적으로 공력데이터베이스는 독립변수가 3개를 초과하지 않으므로 많은 수의 은닉노드가 필요하지 않는다. Training 알고리즘은 Levenberg-Marquardt 방법을 사용하였고 과도하게 fitting되는 것을 방지하기 위해 Bayesian Regularization을 적용하였다.

은닉노드 수에 따른  $R^2$  값의 변화를 Fig. 5에 나타내었다. Fig. 5에서 은닉노드가 최소인 경우에도  $R^2$  값이 0.996 이상임을 확인할 수 있으며 은닉노드의 수가 증가할수록 신경회로망 모델의 추정정확도가 증가함을 확인할 수 있다.

Figure 5에서 은닉노드의 수가 3일 경우부터  $R^2$  값의 증가율이 둔화되므로 함수발생의 정확도를 확보하기 위해서는 최소 3개 이상의 은닉노드를 사용하는 것이 바람직할 것으로 판단된다. 그러나 일반적으로 은닉노드의 수가 증가하면 정확한 데이터 추정을 할 수 있는 반면, 요구되는 계산시간이 증가하기 때문에 은닉노드의 수와 정확도, 계산시간 사이에는 Trade-off가 요구된다.

2.4.2 함수발생 속도 비교 및 분석

시뮬레이터에서 검색시간을 비교하기 위해서 시간간격 동안 함수발생 수를 20, 50, 100, 300, 500, 1,000개로 설정하고 이분검색법, 보간검색법, 각 검색법에 대한 검색창검색법 그리고 3-5개 사이의 은닉노드를 가지는 신경회로망 메타모델을 포함한 총 5개의 기법의 함수발생 속도를 비교하였다. 함수발생 속도를 비교하기 위한 데이터 함수는 식(3), (4)와 같이 생성하였다.

$$\alpha_{test}(n) = \alpha_{test}(n-1) + rand(1) \times (-1)^n \times \alpha_{move} \quad (3)$$

$$M_{test}(n) = M_{test}(n-1) + rand(1) \times (-1)^n \times M_{move} \quad (4)$$

$\alpha_{test}$  와  $M_{test}$  는 초기 시작점에서  $n$ 에 따라 좌우로  $\alpha_{move}$  와  $M_{move}$  만큼 자유롭게 움직이면서 임의의 데이터를 생성할 수 있다. 그리고 검색시간은 계산에 사용되는 컴퓨터의 성능에 따라 달라지기 때문에 각각에 대한 검색속도의 상대적 차이를 확인하기 위해 각 검색법의 평균함수발생 시간을 최대평균 함수 발생시간으로 나누어 비교하였다. 각 은닉노드별 검색속도 결과는 Fig. 6-8에 나타내었다.

Figure 6-8의 결과에서 이분검색법이나 보간검색법 보다 이전의 데이터 패턴을 통해 검색영역을 줄이는 검색창 검색법이 더 빠른 함수발생 속도를 가짐을 확인할 수 있다. 그러나 테스트 독립변수들의 급격한 변화에 따라 더 낮은 함수발생 속도를 가질 가능성도 있다.

Figure 6은 3개의 은닉노드를 사용했을 경우

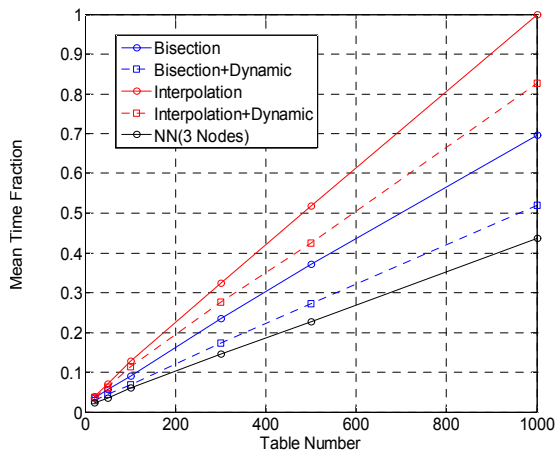


Fig. 6. Mean Time Comparison of Function Generation(Hidden Nodes=3)

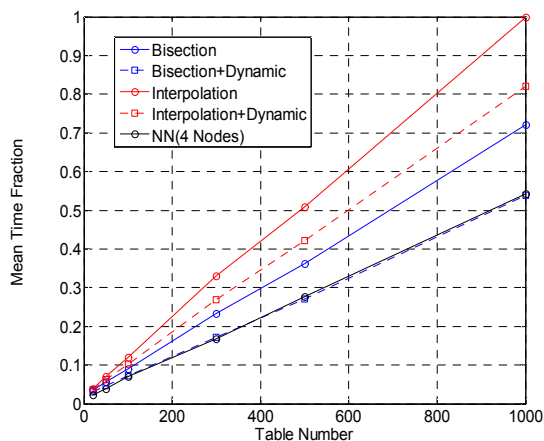


Fig. 7. Mean Time Comparison of Function Generation(Hidden Nodes=4)

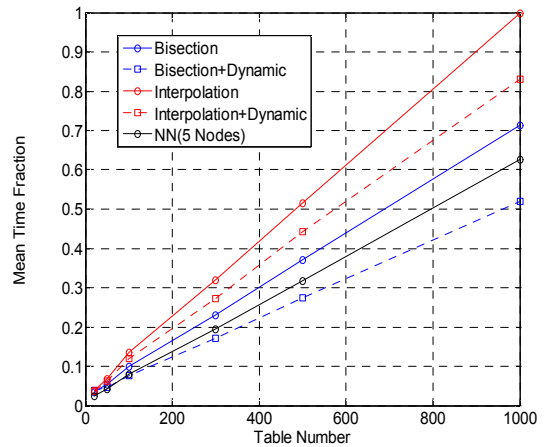


Fig. 8. Mean Time Comparison of Function Generation(Hidden Nodes=5)

신경회로망 메타모델 기법이 테이블 검색 방식의 모든 검색기법에 비해 가장 빠른 검색속도를 가지는 것을 보여주고 있다. 그러나 은닉노드의 수가 4개에서 5개로 증가함에 따라 이분검색법에 검색창 검색법을 결합한 경우에 비해 검색속도가 지연되는 것을 확인할 수 있다. 은닉노드 수의 증가는 데이터 추정정확도를 향상시키지만 반면에 지수함수 계산횟수의 증가에 따라 검색속도가 지연되게 된다. 그러므로 주어진 데이터베이스를 신경회로망 메타모델로 대체하여 함수발생 시간을 감소시키기 위해서는 주어진 데이터베이스의 특성을 파악하여 신경회로망의 은닉모델 수를 데이터 추정 정확도와 함수발생 속도 사이에서 적절한 Trade-off가 요구되어진다.

### 2.4.3 F-16 비선형시뮬레이션을 통한 함수발생 속도비교

실제 F-16 비행시뮬레이션에 제공하는 공력계수  $C_X$ 에 대해 테이블 검색방식과 신경회로망 메

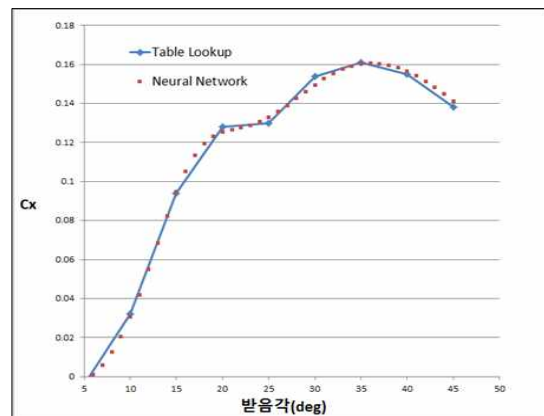


Fig. 9. The Variation of  $C_X$  for  $\alpha$

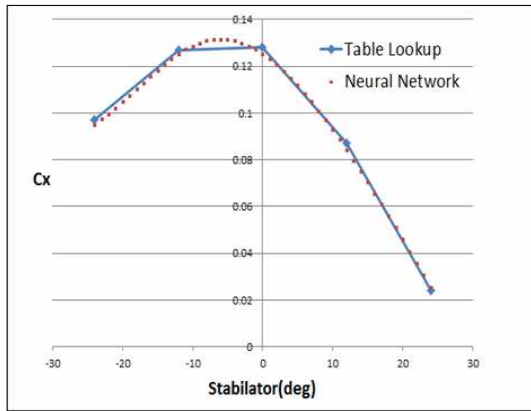


Fig. 10. The Variation of  $C_X$  for Stabilator

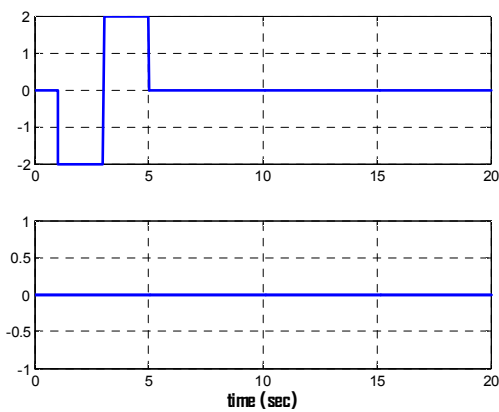


Fig. 11. The Input Forms of Stabilator and Throttle

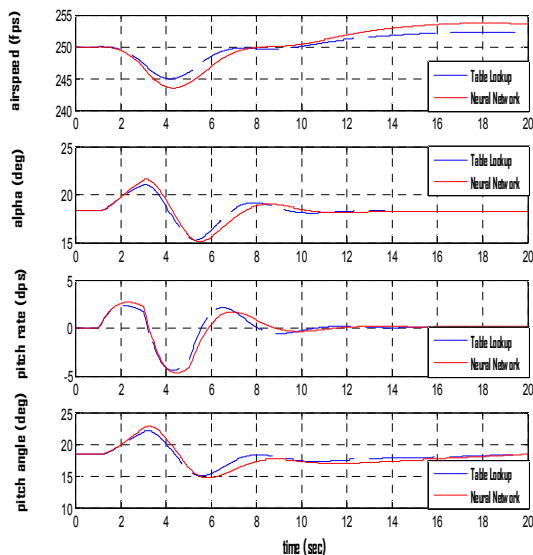


Fig. 12. F-16 Simulation Results (20s)

타모델을 적용하였다. 받음각과 Stabilator에 대한 공력계수  $C_X$ 의 데이터 추정변화를 Fig. 9와 10

에 표시하였다.

F-16 비선형 시뮬레이션 조건은 고도 10,000ft 상공에서 250ft/s로 정상수평비행 상황에서 Fig. 11과 같은 Stabilator와 Throttle에 입력이 가해졌을 때 F-16 항공기의 속도, 받음각, 피치변화율을 20초간 시뮬레이션 한 결과를 Fig. 12에 나타내었다. 각 함수발생 기법에 따른 시뮬레이션 시간은 테이블 검색기법이 0.45초, 5개의 노드를 적용한 신경회로망 메타모델 기법이 0.39초로 계산속도가 13% 정도의 감소를 확인하였으며 본 조건의 시뮬레이션에 대해 실시간성을 향상시킬 수 있음을 알 수 있었다.

Figure 12에서 항공기의 속도에서 테이블 검색 기법과 신경회로망 메타모델 기법 사이에 약간의 오차가 발생하는 것을 확인할 수 있다. 이것은 시뮬레이션에 사용한 F-16 공력모델에 대한 데이터베이스의 큰 비선형성으로 인한 데이터 추정오차로 인해 발생한 것으로 판단되어진다.

데이터 추정정확도는 실측 비행데이터를 통한 오차분석을 통해 확인할 수 있지만 실측 비행데이터 획득이 보안관계상 불가능하였다. 그러나, 비선형영역에서도 선형으로 추정하는 테이블 검색법의 특성상 비선형성이 강할수록 신경회로망 메타모델 기법보다 데이터 추정오차가 클 것으로 판단되어진다. 또한, 신경회로망 메타모델 기법을 적용하면 Fig. 9와 10에서 보이는 것처럼 선형구간 뿐 아니라 비선형성이 강한 구간에서도 데이터의 비선형성을 구현하여 데이터 추정정확도를 향상시킬 수 있으므로 데이터 추정정확도 역시 향상될 것으로 기대된다.

### 2.5 신경회로망의 지수함수 계산속도 향상

신경회로망을 활용한 시뮬레이션에서는 계산시간의 많은 부분이 지수함수에 할당이 된다. 이러한 지수함수의 계산시간을 단축하기 위해 컴퓨터 메모리 구조를 활용한 방법이 제안되었다[4]. 일반적인 IEEE-754(IEEE, 1985)에 규정된 실수(Floating-Point Number)는 컴퓨터에서 64비트의 메모리를 가지며 식(5)와 같이 표현된다.

$$(-1)^s (1+m) 2^{x-x_0} \tag{5}$$

여기서,  $s$ 는 사인 비트이며,  $m$ 은 지수부로서  $[0, 1)$ 사이의 값을 총 52비트로 표현한다.  $x$ 는 지수부로서  $x_0(=1023)$ 의 일정한 바이어스(Bias)를 가진 상태에서 총 11비트로 나타낸다. 일반적인 C 언어의 "union" 변수를 이용하면 동일한 메모리를 두 개 이상의 변수가 공용으로 접근할 수

있다. 즉, 앞쪽 32비트를 정수( $i$ )와 공용으로 접근하면서 정수를 직접적으로 지수승을 수행하고 이를 실수형태로 다시 읽어오는 방식을 취하는 것이다. 만약  $2^y$ 를 구할 때 32비트 정수로 계산하는 방식은 식(6)과 같다.

$$i := 2^{20}(y + 1023) \tag{6}$$

$2^{20}$ 은 앞쪽 32비트에서 실수형식의  $x$ 가 시작되는 곳까지 비트를 이동시키는 역할을 수행한다. 결과적으로  $i$ 가 저장된 동일한 메모리에서 실수변수로 읽어 오면  $2^y$ 의 값을 얻을 수 있다. 구하고자 하는 것은  $e^y$ 이므로  $C = y/\ln(2)$ 로 두게 되면  $2^C = e^y$ 가 되어 식(6)을 이용하기 전에  $y$ 를  $\ln(2)$ 로 먼저 나누어 주면 결과적으로  $e^y$ 를 구할 수 있게 되며 결과적인 식은 식(7)과 같다.

$$i := ay + (b - c) \tag{7}$$

여기서,  $a$ 는  $2^{20}/\ln(2)$ 이며,  $b$ 는  $1023 \cdot 2^{20}$ 이다.  $c$ 는 가수의 일부분인 뒤쪽 32비트가 완전히 무시되므로 정확도를 향상시키기 위한 파라미터 값이다. 본 연구에서는  $c$ 값을 RMS값이 최소가 되도록 60,801로 설정하였다[4].

Figure 13에서는 5개의 은닉노드를 가진 신경회로망 메타모델에 대해 빠른 지수함수 계산 방법을 다른 방법들과 평균함수발생 시간을 비교한 것이다. 본 테스트에서는 일반화를 위해 고성능 전투기의 운용범위인 받음각  $-10^\circ, -6^\circ, -2^\circ, 0^\circ, 1^\circ, 2^\circ, 3^\circ, 5^\circ, 7^\circ, 9^\circ, 11^\circ, 13^\circ, 15^\circ, 17^\circ, 19^\circ, 21^\circ$ 의 16개의 키 값과 마하수의 경우 0.1, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.85, 0.9, 0.95, 1.0, 1.1, 1.15의 13개의 키 값으로 총  $16 \times 13$  크기로 설정하고, 식 (3), (4)에서 급격한 기동을 나타내기 위해 키 값의 임의성을 더 부가하였다. 함수발생수는 100, 500, 1,000, 3,000개로 설정하였다. Fig. 13에서 보듯이 은닉노드가 5개인 경우에도 빠른 지수함수 계산방식이 테이블 검색방식에 비해 함수발생 속도에 대한 성능이 향상됨을 볼 수 있으며 컴퓨터 Math Library에서 제공하는 지수함수 계산 방식에 비해서는 월등함을 알 수 있다.

그러나, 뒤쪽 가수부분을 무시함으로 인해 실제 지수함수 계산에 비해 정확도는 다소 떨어지게 된다. 정확도를 판단하기 위해 표 2에 주어진 데이터에 대해 실제 컴퓨터의 Math Library에서 제공하는 방식으로 계산된 것과 빠른 지수함수 계산방식에 의해 계산된 신경회로망 모델 간의 오차를 분석하였다. Fig. 14는 표 2의 마하수 0.9

에 대해 실제 데이터와 두 가지 방식에 의해 계산된 결과를 비교하여 나타난 것이다. 이 경우에 대해서는 오차가 2.4%이며 전체 데이터에 대해서는 1.6%의 오차가 발생하였다. 추가적으로 주어진 받음각 변수의 중간값에서 테이블검색기법과 비교했을 때의 오차는 기존 지수함수 계산의 경우 6%, 빠른 지수함수 계산 방식의 경우 15%의 오차가 발생하였다.

일반적인 시뮬레이터에서 시뮬레이션 시 테이블 검색에 의한 함수발생이 주어진 데이터베이스의 독립변수에 정확히 일치하는 경우는 드물기 때문에 대부분 선형보간 값에 의해 시뮬레이션이 수행되므로 이러한 오차가 실제와 얼마나 될 것인지는 정확한 비행시험 데이터를 통해 테이블 검색 방법, 신경회로망메타모델에 의한 방법, 빠른 지수계산에 의한 신경회로망 메타모델 계산 방법에 대한 검증이 필요할 것으로 판단된다. 또

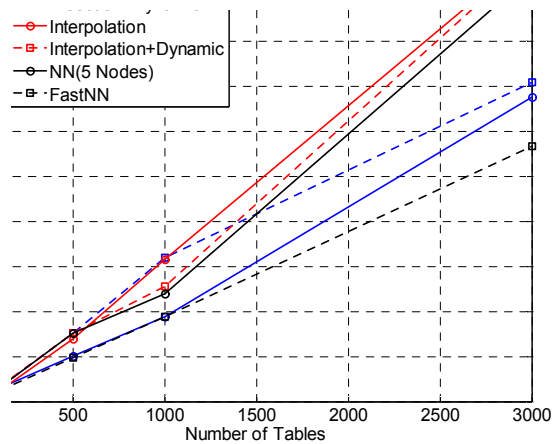


Fig. 13. Mean Time Comparison of Function Generation

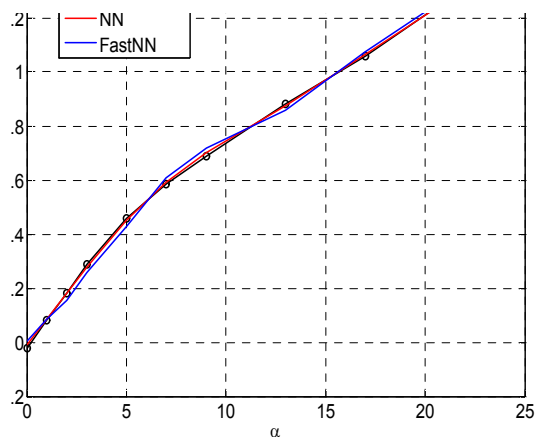


Fig. 14. The Accuracy of Fast Exponential Function Method ( $M=0.9$ )



한, 각 시스템별로 신경회로망 메타모델에서 노드를 감소시키는 것과 빠른 지수계산에 의한 신경회로망 메타모델을 사용하는 것 둘 중 어느 방법이 더 효율적인지에 대한 추가적인 연구가 진행되어야 한다.

### III. 결 론

본 논문에서는 비행 한계점 부근의 비선형적인 공력특성에 대한 구현충실도를 높일 수 있는 가능성을 구현하기 위해 신경회로망 메타모델을 적용하여 시뮬레이터의 데이터베이스를 대체하여 비선형 데이터에 대해 추정 정확도 및 실시간성이 향상될 수 있음을 보였다. 또한, 빠른 지수함수 계산 방법을 적용하여 단일 시뮬레이터 및 향후 LVC 환경에서 함수발생 속도를 높임으로써 구현충실도를 유지하면서 기타 시뮬레이션 큐의 구현충실도 향상을 위한 여유를 제공할 수 있음을 보였다.

그러나, 메타모델로 대체할 시의 정확도와 함수발생 속도를 향상시키는 것은 서로 상충 관계에 있으므로 두 가지를 동시에 향상시킬 수 있는 연구가 향후 추가적으로 필요할 것으로 판단되며, 추정정확도 측면에서 본 논문에서 제시한 것은 주어진 데이터베이스 키 값에 기준한 것이므로 실제와 시뮬레이션의 정확도를 판단하기 위해서는 비행시험 데이터와의 비교, 분석이 요구된다.

공력특성에 대한 구현충실도를 유지하면서 빠른 함수발생은 시뮬레이터의 모의 갱신율을 높이고 전체 시뮬레이터의 지연시간 감소라는 잇점이 있다. 특히, 향후 LVC 연동 시 주어진 시간간격 내에 다수 개체간 통신, 개체간 해상도 변환, 다수 객체의 충실도 높은 구현에 많은 도움이 될 것으로 예상된다.

### 후 기

본 연구는 국방과학연구소의 지원(UD120 060JD)에 의해 수행되었으며, 이에 감사드립니다.

### References

- 1) John Croft, "Amping the Envelope", Aviation Week & Space Technology, Dec. 3, 2012.
- 2) William J. Bezdek, Joel Maleport, Robert Z. Olshan, "Live, Virtual & Constructive Simulation for Real Time Rapid Prototyping, Experimentation and Testing using Network Centric Operations", AIAA Modeling and Simulation Technologies Conference and Exhibit, AIAA-2008-7090, 2008.
- 3) Guoqiang Zhang, B. Eddy Patuwo, Michael Y. Hu, "Forecasting with Artificial Neural Networks: The State of the Art", International Journal of Forecasting, Vol. 14, 1998, pp. 35-62.
- 4) N. N. Schraudolph, "A Fast, Compact Approximation of the Exponential Function", Neural Computation, Vol.11, No.4, 1999, pp. 853-862.
- 5) Yoon S. and Kang H., "Dynamic-Window Search for Real-time Simulation of Dynamic Systems", KSAS, Vol.28, No.4, 2000.6, pp. 86-92.
- 6) Yoon S. and Kang H., "A Study on Searching Speeds of Real-time Searching Algorithms", KSAS, Vol.28, No.1, 2000.2, pp. 126-132.
- 7) Peter H. Zipfel, Modeling and Simulation of Aerospace Vehicle Dynamics, AIAA Education Series, 2007.