

<학술논문>

DOI <https://doi.org/10.3795/KSME-A.2021.45.10.853>

ISSN 1226-4873(Print)

2288-5226(Online)

유연 다물체 동역학 시스템의 실시간 해석을 위한 DNN 기반 메타모델링 기법[§]

한성지* · 최희선** · 최주환*** · 최진환* · 김진균*[†]

* 경희대학교 기계공학과, ** 콜로라도대학교 볼더, *** 펑션베이(주)

A DNN-based Metamodeling Techniques for Real-Time Simulations of Flexible Multibody System Dynamics

Seongji Han*, Hee-Sun Choi**, Juhwan Choi***, Jin Hwan Choi* and Jin-Gyun Kim*[†]

* Dept. of Mechanical Engineering (Integrated Engineering), Kyung Hee Univ.,

** University of Colorado, Boulder, *** FunctionBay, Inc.

(Received March 3, 2021 ; Revised May 12, 2021 ; Accepted May 25, 2021)

Key Words: Flexible Multibody Dynamics(유연 다물체 동역학), Deep Neural Networks(심층신경망), Stochastic Gradient Descent(확률적 경사 하강), Error Correction(오차 보정), Real-Time Simulation(실시간 시뮬레이션)

초록: 본 연구에서는 유연 다물체 동역학 시스템에 포함된 유연체의 절점 위치, 응력 및 변형률을 지정된 설계 변수 입력을 통해 실시간으로 얻기 위한 심층신경망 기반 메타모델링 기법을 제시하였다. 유연체의 거동을 학습시키기 위해 적절한 형태의 입력 및 출력 데이터 구조를 구성하였으며, 수많은 유연체 절점 자유도로 인한 데이터 증가로 발생하는 학습 시간을 단축하기 위한 효율적인 훈련 알고리즘을 개발하였다. 알고리즘은 확률적 경사 하강(SGD: stochastic gradient descent) 단계와 오차 보정 단계로 구성되며, 전체 데이터의 일부만을 사용하여 학습된다. 확률적 경사 하강 단계에서 심층신경망은 거친 데이터 모음(coarse data set)을 사용하여 순차적/반복적으로 학습되며, 이 과정에서 심층신경망 모델의 개선이 충분하지 않은 경우 오차 보정 단계를 통해 정확도를 추가적으로 향상시킬 수 있다. 개발된 알고리즘의 효율성 및 정확성, 실시간성을 수치 예제를 통해 확인하였다.

Abstract: In this study, a deep neural network-based metamodeling technique is proposed to obtain the positions, stresses, and strains of the nodes of a flexible body in real-time using the input of designated design variables for the flexible multibody dynamics systems. To train the responses of a flexible body, an appropriate type of input and output data structure is constructed, and an efficient training algorithm is developed to reduce the computational resources caused by data increase owing to the numerous degrees of freedom of nodes in the flexible body. The algorithm includes a stochastic gradient descent (SGD) step and an error correction step, and the deep neural network model is trained using only a part of the entire data. In the SGD step, the model is trained sequentially and iteratively using a coarse data set. As the iterations proceed, when the improvement of the model is insufficient, the accuracy is further improved by an error correction step. The efficiency, accuracy, and the possibility of real-time simulation of the proposed algorithm are presented through a numerical example.

- 기호설명 -

§ 이 논문은 대한기계학회 2020년 학술대회(2020. 12. 16.-24., 온라인) 발표 논문임.

[†] Corresponding Author, jingyun.kim@khu.ac.kr

© 2021 The Korean Society of Mechanical Engineers

I : 입력 데이터

O : 출력 데이터

\mathcal{N} : 심층신경망 모델

\mathcal{N}^{SGD}	: 확률적 경사 하강 모델
\mathcal{N}^{EC}	: 오차 보정 모델
\mathcal{M}	: 기준 모델
n_x	: 유연체 절점의 수
t	: 시물레이션 시간
t_{end}	: 시물레이션 종료 시간
x, y, z	: 절점의 초기 위치
$\tilde{x}, \tilde{y}, \tilde{z}$: 시간에 대한 절점의 위치
E	: 탄성계수
ρ	: 밀도
ν	: 포와송 비
σ	: 본-미세스 응력
ε	: 변형률
\mathbf{F}	: 전체 데이터
\mathbf{C}_k	: 거친 데이터 모음
\mathbf{V}	: 검증 데이터 모음
\mathbf{E}^{EC}	: 오차 보정 데이터 모음
θ, η	: 심층신경망 모델의 가중치 및 편향
\mathcal{R}^2	: 결정계수
\mathbf{h}	: $\mathbf{y} = \mathbf{x}$ 그래프로부터 샘플의 거리
ρ_m	: 중량물의 질량
θ_r	: 상부체 회전각

1. 서 론

유연 다물체 동역학(FMBD: flexible multibody dynamics) 시스템의 실시간 시물레이션은 다양한 산업 분야에서 중요한 이슈이며,^(1,2) 해석 시간 단축을 위해 다양한 차수 축소 모델링(ROM: reduced order modeling) 기법⁽³⁻⁶⁾이 개발되어 사용되고 있다. 하지만 이러한 축소 모델링 기법들은 유연체에 대한 여러 가지 가정에 기반하고 있기 때문에, 높은 비선형성을 가지는 유연 다물체 동역학 시스템을 표현하는데 다양한 오차가 발생하게 된다. 또한 축소된 자유도만큼의 시스템 운동방정식을 순차적으로 풀어야 하기 때문에 실시간 해석을 위해 사용되기에는 한계가 있다. 이를 해결하기 위해 시스템의 비선형성을 효과적으로 묘사할 수 있는 심층신경망(DNN: deep neural networks)을 활용한 메타 모델링 기법이 개발되었다.⁽⁷⁻¹⁰⁾ 이는 과거 다양한 종류의 신경망이 입력된 데이터의 분류,⁽¹¹⁻¹⁵⁾ 신호의 필터링^(13,16) 등 신호처리의 관점에서 연구되어 온 것과는 달리, 본 연구에서는 심층신경망을 사용한 메타 모델 개발을 통해 비선형 수치해석 모델의 동적응답을 실시간으로 예측하는 것에 초점이 맞춰져 있다. 이때 유연 다물체 동역학 모델의 특성상 필연적인 수많은 학습 데이터 문제를 해결

하기 위해 이 중 적은 수의 일부 데이터만을 사용하여 전체 데이터를 모두 사용했을 때의 심층신경망 모델과 유사한 정확도를 보이는 효율적인 기법을 제시한다.^(9,10)

신경망 학습에서의 효율적인 데이터 사용에 대한 연구는 주로 스마트 샘플링^(17,18)이라는 주제로 연구되며, 기준 모델의 해가 있는 경우에 대해서 선택된 데이터를 전부 활용하는 것이 아닌, 반복 및 오차 보정을 활용하여 점진적으로 사용하는 데이터 양을 늘려가며 매 반복의 모델 평가를 통해 필요한 만큼의 데이터만을 사용하는 방식을 제안하였다.

제안된 알고리즘은 확률적 경사 하강(SGD: stochastic gradient descent) 단계와 오차 보정(EC: error correction) 총 두 가지 단계로 구성되어 있으며, 전체 데이터를 사용한 것과 유사한 성능을 보이는 것을 확인하였다.

2. 학습 데이터 구조

심층신경망을 활용한 메타모델링의 목표는 시스템에 영향을 주는 임의의 입력 변수에 따른 동적응답을 실시간으로 도출해주는 것이다. 이때 입력 변수와 동적 응답은 각각 심층신경망 모델의 입력 변수 \mathcal{I} 와 출력 변수 \mathcal{O} 가 된다. 시스템을 효과적으로 묘사하기 위해 적절한 입력 변수 및 출력 변수를 선정하는 것이 필수적이며, 학습에 적합한 데이터 구조를 구축해야 한다. 학습된 심층신경망 모델 \mathcal{N} 은 입력 변수와 출력 변수의 함수로 $\mathcal{O} = \mathcal{N}(\mathcal{I})$ 와 같이 표현될 수 있다.

2.1 입력 변수

필수적인 입력 변수들은 공간 입력 및 시간 입력이 있다. 유연체는 수많은 절점으로 구성되며 심층신경망 모델에 해당 절점의 정보를 입력해 주어야 한다. 이는 각 절점의 x, y, z 초기 위치를 의미하며 $\mathcal{J}_1, \mathcal{J}_2, \mathcal{J}_3$ 으로 정의한다. 또한 원하는 시점의 출력을 도출하기 위해 시물레이션 종료 시간까지 이산화된 시간 입력 \mathcal{J}_4 이 사용되어야 한다. 필수적인 입력 변수는 식 (1)과 같이 정의되며 이때 n_x 과 t_{end} 는 각각 절점의 수, 시물레이션 종료 시간을 의미한다.

$$\begin{aligned} \mathcal{J}_1, \mathcal{J}_2, \mathcal{J}_3 &= x_i, y_i, z_i, \quad i = 1 \cdots n_x, \\ \mathcal{J}_4 &= t, t \in [0, t_{end}] \end{aligned} \quad (1)$$

설계 변수 $\mathcal{J}_i, i = 5 \cdots$ 의 경우 시스템 응답을 변화시키는 모든 종류의 변수가 될 수 있다. 유연체

의 물성, 입력되는 외력, 부상 혹은 스프링의 강성 및 감쇠 등이 선택될 수 있으며 시스템의 해를 관측하고자 하는 범위 내에서 선정된다. 유연체의 탄성계수 E 와 밀도 ρ 가 설계 변수로 선정된 경우, 식 (2)와 같이 표현할 수 있다.

$$J_4 = E, I_5 = \rho \quad (2)$$

2.2 출력 변수

시스템에서 측정 가능한 모든 응답은 출력 변수로 선정될 수 있다. 본 연구에서는 시간에 대한 유연체의 절점의 위치 $\tilde{x}, \tilde{y}, \tilde{z}$ 및 본-미세스 응력 σ 및 변형률 ϵ 을 출력 변수로 선정하여 심층신경망을 활용한 실시간 응답 도출이 가능하도록 하였다. 이와 같이 데이터 구조를 생성한 경우 심층신경망 모델은 Fig. 1과 같이 구성된다.

3. 확률적 경사 하강-오차 보정 알고리즘

해석의 정확도를 위해 유연체는 많은 수의 절점을 포함하기 때문에 심층신경망 학습을 위해 필요한 데이터는 절점 수에 비례하여 증가하게 된다. 심층신경망 모델의 학습 시간 단축을 위해 일부의 거친 데이터 모음 \mathbf{C}_k 를 활용하여 순차적으로 모델을 학습하는 확률적 경사 하강 단계와, 모델의 정확도를 높이기 위한 오차 보정이 융합된 알고리즘을 제시한다.

3.1 확률적 경사 하강 단계

확률적 경사 하강 단계에서는 전체 데이터 \mathbf{F} 중 일부의 데이터를 선택하여 거친 데이터 모음 $\mathbf{C}_k (k = 1, \dots)$ 를 구성한다. 여기서 하첨자 k 는 k 번째 거친 데이터 모음을 의미하며 전체 데이터 중 매우 적은 양의 데이터가 임의로 선택될 수 있다.

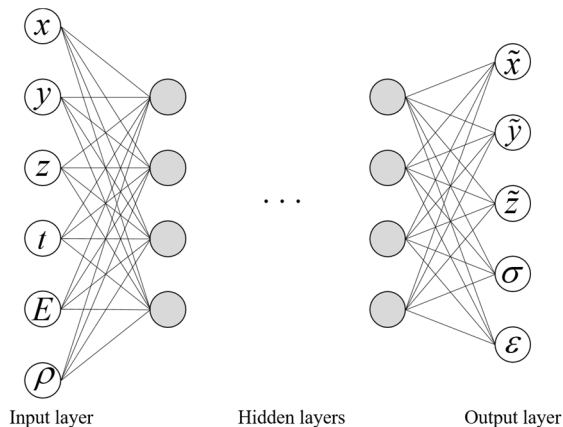


Fig. 1 Structure of a DNN model

가중치(weight)와 편향(bias)이 θ_{k-1} 로 정의된 심층신경망 모델 \mathcal{N}^{k-1} 은 식 (3)과 같이 정의된다.

$$\mathcal{N}^{k-1} = \mathcal{N}[\theta_{k-1}] \quad (3)$$

심층신경망 모델 \mathcal{N}^{k-1} 을 \mathbf{C}_k 를 사용해 학습시켜 가중치와 편향이 갱신된 모델 \mathcal{N}^k 을 식 (3)과 같이 얻을 수 있다.

$$\mathcal{N}^k = \mathcal{N}[\theta_k; \theta_{k-1}, \mathbf{C}_k] \quad (4)$$

학습된 k 번째 모델을 평가하는 방식으로 검증 데이터 모음 \mathbf{V} 를 활용한다. 검증 데이터 모음은 전체 데이터 중 무작위로 선정된 약 10%의 데이터로 구성하였으며, 학습에 사용되는 \mathbf{C}_k 와 중복되지 않도록 선택하였다. 데이터들의 관계는 Fig. 2에서 확인할 수 있다.

모델 평가 및 반복 여부 판단 과정은 다음과 같다. 검증 데이터 모음에 대한 심층신경망 모델의 출력과 기준 모델의 출력의 결정계수 \mathcal{R}^2 를 계산한다. 모든 출력에 대한 \mathcal{R}^2 의 평균값을 최근 5번의 확률적 경사 하강 반복 동안 추적하며, 5번의 \mathcal{R}^2 의 평균값을 선형 보간하여 얻은 직선의 기울기의 절대값 $|\mathbf{s}_{SGD}(k)|$ 과 선정한 공차 tol_s 의 크기를 비교한다.

만약 $|\mathbf{s}_{SGD}(k)|$ 가 tol_s 보다 작은 경우 확률적 경사 하강 단계를 통한 모델 개선이 미미하다고 판

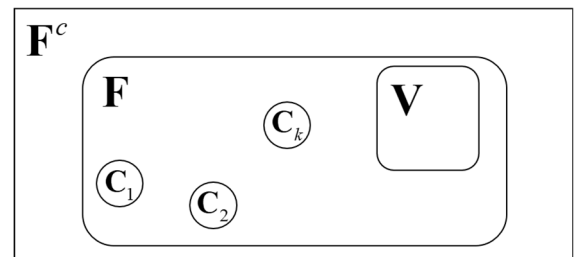


Fig. 2 Diagram of the relationship of data set

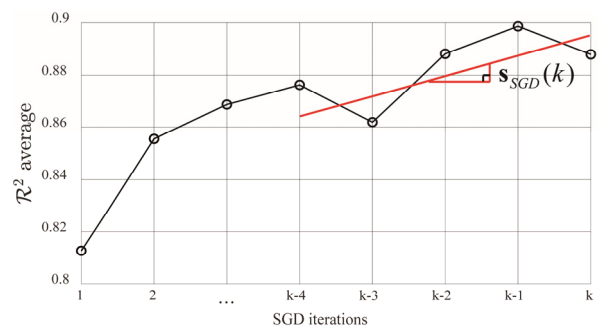


Fig. 3 A linear regression of the average of R-squared value and its slope $\mathbf{s}_{SGD}(k)$

단하고 오차 보정 단계로 넘어갈 수 있도록 하였다. 이 과정을 Fig. 3에 나타내었다.

확률적 경사 하강 단계를 통해 최종적으로 심층신경망 모델 \mathcal{N}^{SGD} 를 얻을 수 있다. 확률적 경사 하강 모델 \mathcal{N}^{SGD} 는 충분히 정확한 결과를 도출할 수 있지만 그렇지 않을 수도 있다. 따라서 \mathcal{N}^{SGD} 모델의 정확성을 개선하기 위한 오차 보정 단계가 도입된다.

3.2 오차 보정

확률적 경사 하강 단계에서 얻은 심층신경망 모델의 오차를 보정하기 위해 \mathcal{N}^{SGD} 의 오차 함수인 새로운 심층신경망 모델 \mathcal{N}^{EC} 가 도입된다. 시뮬레이션 해석 결과를 도출하는 기준 모델을 \mathcal{M} 이라고 가정하면, 입력 변수 \mathcal{I}^{EC} 에 대한 \mathcal{N}^{SGD} 의 오차 \mathcal{O}^{EC} 는 식 (5)와 같이 표현할 수 있다.

$$\mathcal{O}^{EC} = \mathcal{M}(\mathcal{I}^{EC}) - \mathcal{N}(\mathcal{I}^{EC}) \quad (5)$$

이 오차 \mathcal{O}^{EC} 를 근사하는 모델 \mathcal{N}^{EC} 를 얻을 수 있다면 \mathcal{N}^{SGD} 와 \mathcal{N}^{EC} 를 활용하여 모델 \mathcal{M} 과 유사한 결과를 식 (6)과 같이 얻을 수 있을 것이다.

$$\mathcal{M} \approx \mathcal{N}^{SGD-EC} = \mathcal{N}^{SGD} + \mathcal{N}^{EC} \quad (6)$$

\mathcal{N}^{EC} 를 학습시키는 과정은 다음과 같다. 먼저 가중치와 편향 η_0 로 구성된 심층신경망 모델 $\mathcal{N}^{EC}[\eta_0]$ 를 초기화한다.

검증 데이터 모음의 입력과 출력 $\mathcal{I}^V, \mathcal{O}^V$ 를 활용하여 심층신경망 모델의 출력에 대한 레이블(label)대 예측(prediction)을 Fig. 4와 같이 얻을 수 있다.

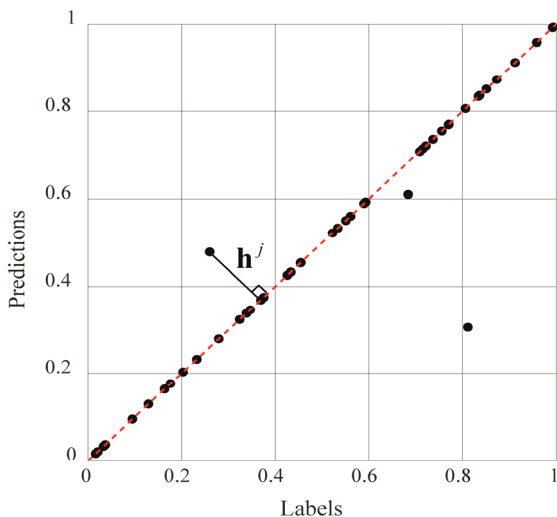


Fig. 4 The distance between the line $y = x$ and samples

이때 $y = x$ 그래프로부터 데이터 모음 중 임의의 표본과의 거리 \mathbf{h} 를 식 (7)과 같이 계산하여 기준 모델 \mathcal{M} 에서 도출되는 해와 차이가 큰 표본들을 판단할 수 있다. 이때 j 는 j 번째 출력 변수를 의미한다.

$$\mathbf{h}^j(\mathcal{I}^V) := \frac{1}{\sqrt{2}} |\mathcal{O}^V - \mathcal{N}_j^{SGD}(\mathcal{I}^V)| \quad (7)$$

표본 i 에 대한 $y = x$ 로부터의 거리를 식 (8)과 같이 계산한 후 모델 \mathcal{M} 과의 오차가 큰 표본들을 선택한다.

$$\mathbf{h}_i(\mathcal{I}^V) := \|\mathbf{h}_i^x, \mathbf{h}_i^y, \dots, \mathbf{h}_i^l\| \quad (8)$$

전체 검증 데이터 모음에 대한 표본들의 거리 \mathbf{h} 의 평균 $tol_{\mathbf{h}}$ 를 계산하여 이보다 큰 거리를 가지는 표본들의 입력 변수들로 오차 보정을 위한 새로운 입력 변수 \mathcal{I}^{EC} 를 식 (9)와 같이 구성할 수 있다. \mathcal{I}^{EC} 를 확률적 경사 하강 모델에 입력할 때 얻는 출력과 기준 모델의 해의 차이를 식 (10)과 같이 도출하여 오차 보정을 위한 출력 변수 \mathcal{O}^{EC} 로 선정한다. 오차 보정을 위한 데이터 모음 \mathbf{E}^{EC} 는 식 (11)과 같이 정의된다.

Inputs:

1: Get full data from simulations $\mathbf{F} := \{\mathcal{I}, \mathcal{O}\}$

SGD step:

2: Initialize a DNN model

$\mathcal{N}^0 := \mathcal{N}[\theta_0]$

3: Choose random sets of coarse data \mathbf{C}_k

4: Define validation set \mathbf{V}

5: **while** $s_{SGD}(k) > tol_s$ **do**

6: Train \mathcal{N} with \mathbf{C}_k

7: Get $\mathcal{N}^k := \mathcal{N}[\theta_k; \theta_{k-1}, \mathbf{C}_k]$

8: Calculate $\mathcal{R}^2(\mathcal{N}^k)$ and $s_{SGD}(k)$

9: **end while**

10: Get DNN model \mathcal{N}^{SGD}

EC step:

11: Initialize another DNN model

$\mathcal{N}^{EC}[\eta_0]$

12: Define the distance \mathbf{h} that measures error of samples

13: Find new input \mathcal{I}^{EC} from \mathbf{V} such that

$\mathcal{I}^{EC} := \{\mathcal{I}^* \in \mathcal{I}^V : \mathbf{h}(\mathcal{I}^*) > tol_{\mathbf{h}}\}$

14: Construct training data

\mathbf{E}^{EC} for \mathcal{N}^{EC} : $\mathbf{E}^{EC} = \{\mathcal{I}^{EC}, \mathcal{M}(\mathcal{I}^{EC}) - \mathcal{N}^{SGD}(\mathcal{I}^{EC})\}$

15: Train \mathcal{N}^{EC} with \mathbf{E}^{EC}

16: Get DNN model \mathcal{N}^{EC}

Output:

17: $\mathcal{N}^{SGD-EC} = \mathcal{N}^{SGD} + \mathcal{N}^{EC}$

Fig. 5 Flowchart of the SGD-EC algorithm

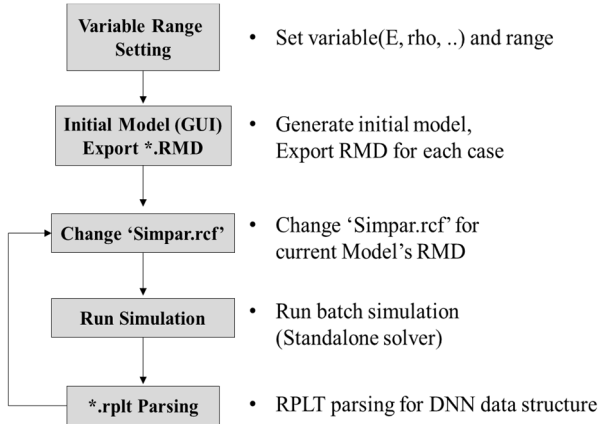


Fig. 6 Schematic diagram of automated process to collect training data

$$J^{EC} := J^* \in \mathcal{J}^V: \mathbf{h}(J^*) > tol_h \quad (9)$$

$$\mathcal{O}^{EC} := \mathcal{M}_j(J^{EC}) - \mathcal{N}_j^{SGD}(J^{EC}) \quad (10)$$

$$\mathbf{E}^{EC} = [J^{EC} \mathcal{O}^{EC}] \quad (11)$$

\mathbf{E}^{EC} 를 활용하여 모델 \mathcal{N}^{EC} 를 학습시키면, 식 (12)와 같은 오차 보정 모델을 얻을 수 있다. Fig. 5는 제안된 기법의 알고리즘을 정리한 것이다.

$$\mathcal{N}^{EC} = \mathcal{N}[\eta; \eta_0, \mathbf{E}^{EC}] \quad (12)$$

3.3 자동화 프로세스

심층신경망 학습을 위해 충분한 양의 데이터를 취득해야 한다. 본 연구에서는 학습 데이터로 활용될 해석 결과 추출을 위해 상용 다물체 동역학 프로그램 RecurDyn을 사용하였으며, 반복적인 모델링 및 해석, 결과 가공 등을 C# 언어로 프로그래밍할 수 있는 내장 기능인 ProcessNet을 활용하여 데이터 취득 과정을 자동화하였다.⁽¹²⁾

먼저 설계 변수의 범위를 선정하고, 해석을 위한 초기 모델을 생성한다. 가능한 모든 설계 변수의 조합에 대한 해석 모델 정보 파일(*.rmd)을 추출하여 해당 모델에 대한 시뮬레이션 변수 파일 'Simpar.rcf'를 연결한다. 배치 명령어를 사용하여 필요한 만큼의 해석을 수행하고, 해석 결과 파일(*.rplt)을 심층신경망 학습에 맞는 데이터 구조로 변환하는 작업을 자동화 할 수 있도록 하였다. 이 과정을 나타낸 개략도는 Fig. 6과 같다.

4. 수치 예제

제안된 학습 기법의 성능을 확인하기 위해 Fig. 7과 같은 수치 예제⁽⁹⁻¹¹⁾를 사용하였다. 제시된 예제는 굴삭기 시스템으로 RecurDyn에서 제공하는

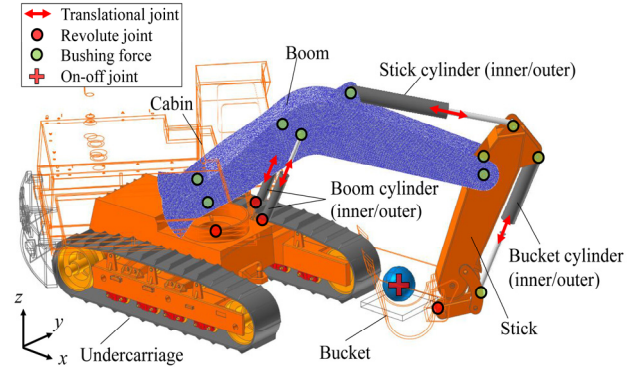


Fig. 7 Configuration of excavator model

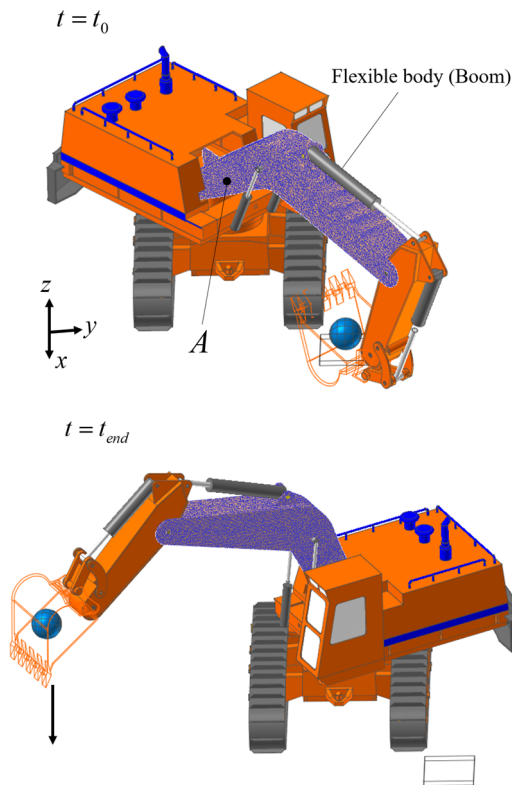
예제 모델을 수정하여 구축하였다.⁽¹²⁾ 유연 다물체 동역학 시스템의 요소인 조인트, 힘 등으로 구성되며 충분히 많은 자유도를 갖는 유연체가 포함되어 일반적인 유연 다물체 동역학 해석을 묘사하는데 적합한 것으로 판단된다.

굴삭기의 언더캐리지는 지면에 고정되어 구속된 것으로 가정하며, 언더캐리지와 캐빈은 회전조인트로 연결되어 상부체의 z축방향 회전 변위를 인가할 수 있도록 구성되어 있다. 캐빈은 붐의 하단부 양쪽 면과 부싱 힘으로 연결되며 붐의 옆면에서 수직인 축을 제외한 자유도에 대해 강성 및 감쇠계수로 붐을 지지한다. 캐빈과 좌, 우측 붐 실린더는 회전조인트로 연결되었으며 각각의 붐 실린더는 내부와 외부 강체로 구성되어 병진조인트를 통해 붐의 상하운동을 묘사할 수 있도록 한다. 붐 실린더의 상단부는 붐의 절곡부 중심과 부싱 힘으로 연결되며, 붐과 스틱을 연결하는 스틱 실린더도 동일한 형태로 모델링되어 있다. 스틱은 붐과 붐 실린더에 대해 부싱 힘으로 연결되며, 버킷 실린더를 통해 버킷의 개폐를 제어할 수 있도록 한다. 버킷과 구형 강체로 표현된 중량물은 정해진 조건에 따라 특정 자유도를 구속하거나 풀어줄 수 있는 온-오프 조인트로 구속되어 있으며, 조건을 만족하기 전까지 모든 자유도를 구속하며 조건을 만족한 이후에는 모든 자유도를 풀어주어 특정 시점에서 굴삭기가 중량물을 떨어뜨리는 거동을 묘사할 수 있도록 하였다. 여기서 붐은 유연체로 모델링하였으며 384,074개의 사면체 요소로 구성되었다. 붐을 제외한 모든 부품은 강체로 설정하였다.

굴삭기의 동작 과정은 Fig. 8에서 확인할 수 있다. 초기 상태 t_0 에서 굴삭기는 중량물을 든 채 정지해 있다. 시뮬레이션이 시작되면 굴삭기는 중량물을 정해진 높이까지 들어올리고, 회전각 θ_r 만큼 회전하여 붐을 펼쳐 중량물을 떨어뜨리게 된다.

Table 1 Fixed constants and design variables

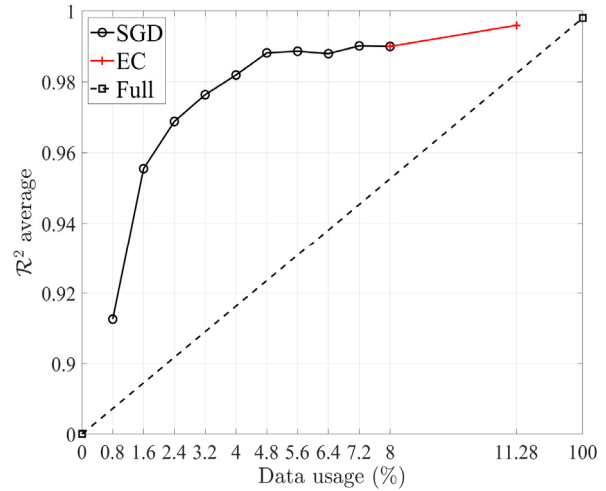
Parameters	Value & ranges	Mesh size
ρ (kg/mm ³)	$7.85e^{-6}$	-
ν	0.28	-
E (GPa)	[150, 250]	$\Delta E = 11.11$
ρ_m (kg/mm ³)	[$3.93e^{-6}$, $7.85e^{-6}$]	$\Delta \rho_m = 4.356e^{-7}$
θ_r (rad)	[0.5236, 1.0472]	$\Delta \theta_r = 0.0582$

**Fig. 8** The operation of the excavator model

유연체의 밀도 ρ 와 포와송 비 ν 는 고정 변수로 선정되었으며, 탄성계수 E 와 중량물의 질량 ρ_m , 회전각 θ_r 가 설계 변수로 선정되었다. 고정 변수와 설계 변수는 각각 Table 1과 같은 값 및 범위를 가지며 설계 변수의 경우 지정된 범위 내에서 균등한 간격으로 분할되었다.

제안된 기법의 효율성을 확인하기 위해 사용된 데이터 양에 따른 정확도를 Fig. 9와 같이 비교하였다. 범례에 표기된 ‘full’은 전체 데이터를 사용하여 학습된 심층신경망 모델을 의미하며, ‘SGD’와 ‘EC’는 각각 확률적 경사 하강 모델 및 오차 보정 모델을 의미한다.

정확도를 비교하기 위한 지표로는 심층신경망 모델의 출력 변수들의 \mathcal{R}^2 평균을 사용하였다. 한번의 확률적 경사 하강 단계에는 전체 데이터의 0.8%가 사용되며, 오차 보정 단계로 도입하기 전

**Fig. 9** Data usage comparisons

까지 10번의 반복이 진행된다. 반복 과정에서 최근 5번의 \mathcal{R}^2 평균값을 추적하여 모델의 개선 여부를 판단하고, $|s_{SGD}(k)|$ 가 tol_s 보다 작아지는 10번째 반복에서 오차 보정 단계로 진입하게 된다.

‘SGD’ 모델의 정확도가 오차 보정 모델 ‘EC’를 통해 개선되는 것을 확인할 수 있으며 단계가 진행됨에 따라 누적되는 데이터의 양을 x축에 나타내었다. ‘Full’ 모델 점선 그래프는 한 번에 모든 데이터를 활용하여 학습된 \mathcal{R}^2 평균이 1에 가까운 값을 갖는 것을 보여준다.

오차 보정 단계에서 사용된 데이터는 약 3.28%로, 제안된 알고리즘에서 사용한 데이터 총량은 약 11.28%이다. 이를 통해 약 11.28%의 데이터만을 사용하여 전체 데이터를 사용하여 학습된 ‘full’ 모델과 유사한 정확도를 얻을 수 있음을 확인하였다.

학습된 심층신경망 모델이 도출하는 출력 변수의 시간 영역에서 적절성을 확인하기 위해 Fig. 8의 절점 A에 대한 응답을 Fig. 10과 11에서 나타내었다. 학습된 범위에 포함되는 임의의 설계 변수는 식 (13)의 값을 사용하였다.

$$\begin{aligned} E &= 215.51(\text{GPa}) \\ \rho_m &= 6.08e^{-5}(\text{kg/mm}^3) \\ \theta_r &= 0.838(\text{rad}) \end{aligned} \quad (13)$$

Fig. 10은 순서대로 시간에 대한 절점 A의 x, y, z 위치가 보여주지며, Fig. 11은 각각 본-미세스 응력 및 변형률을 의미한다. 식 (12)와 동일한 설계변수를 사용하여 RecurDyn 프로그램으로부터 얻은 수치 해석 결과를 기준 모델 ‘reference’로 선정하였으며 학습된 심층신경망 모델 ‘DNN’이 예측하는 출력 변수의 시간에 따른 응답이 기준 모델과 유사한

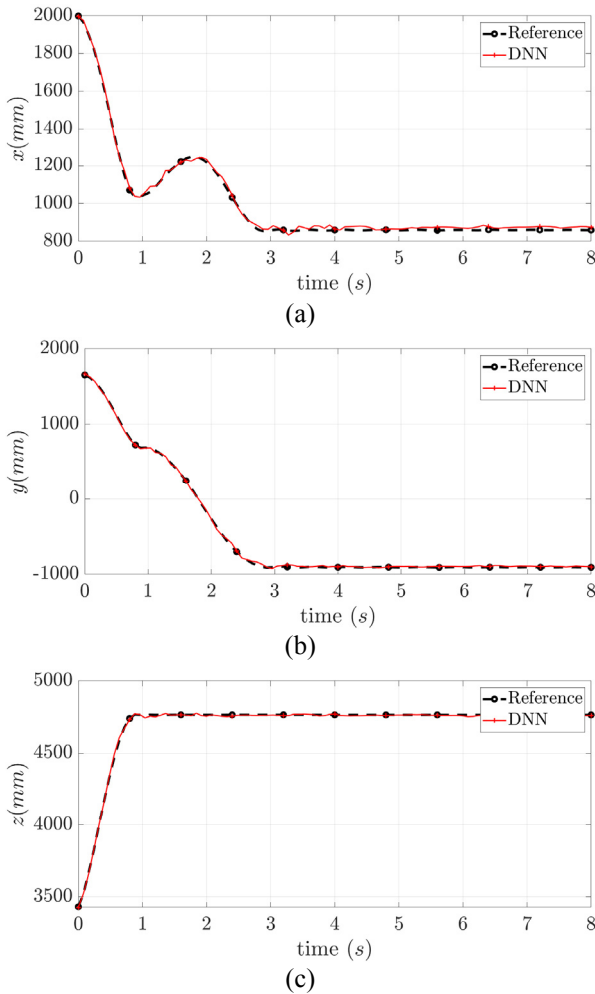


Fig. 10 Time transient position comparisons

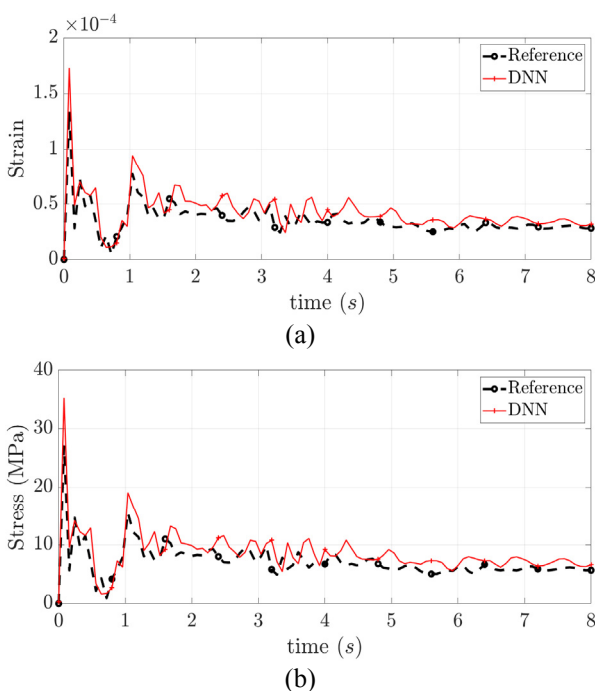


Fig. 11 Stress and strain comparisons

Table 2 CPU time comparisons for the number of nodes

Case	CPU time (ms)
Reference	326.6
10 nodes	4.018
100 nodes	8.147
500 nodes	22.247
1,000 nodes	43.757

것을 확인할 수 있다.

Table 2에서는 하나의 시간 간격을 푸는데 소요되는 시간에 대해 기준 모델과 심층신경망 모델을 비교하였다. 심층신경망의 경우 원하는 절점 수만큼의 정보만을 선택적으로 얻을 수 있기 때문에 절점 10, 100, 500, 1,000개에 대한 결과 도출의 소요 시간을 측정하였으며, 각각의 결과는 100번을 테스트하여 그 평균값을 제시하였다.

전체 시스템의 운동 방정식을 풀어야 하는 기준 모델과는 다르게 심층신경망 모델로 절점 10개의 결과를 얻는 시간은 약 81배 빠른 것을 확인할 수 있다.

본 연구에서 제시하는 메타모델링 기법 및 학습 알고리즘은 특정한 모델에 종속되는 것이 아닌 일반적인 형태의 유연 다물체 동역학 시스템에 대하여 적용될 수 있으며 시스템에 따라 다양한 입력 및 출력 변수가 사용될 수 있다.⁽⁹⁾

5. 결 론

본 연구에서는 FMBD 시스템의 실시간 해석을 위한 심층신경망 메타모델링 기법 및 효율적인 학습 알고리즘을 제시하였다. 수치 예제를 통해 전체 데이터 중 일부 데이터만을 사용하여 학습시킨 심층신경망 모델이 전체 데이터를 사용하여 학습된 모델의 결과와 유사하며, 기준 해의 정확성을 충분히 만족시키며 실시간으로 해석 결과를 도출해주는 것을 확인하였다.

제안된 알고리즘을 활용하여 다양한 심층신경망 모델 학습에 필요한 학습 시간이 단축될 수 있을 것으로 기대된다.

개발된 알고리즘에서 사용되는 입력 변수에는 시간 입력이 반드시 포함되어야 한다. 이는 생성된 메타 모델이 학습에 사용된 수치 모델의 시물레이션 종료 시간 이후의 결과에 대한 정확성을 보장받지 못한다는 것을 의미한다. 실시간 제어나 실제 시스템의 실시간 시물레이션의 관점에서 시스템의 종료 시간이 한정되거나 일정한 작동 상황

이 반복되는 경우에는 본 알고리즘을 활용한 메타 모델링이 효율적이나, 종료 시간이 정해지지 않은 시스템에 대해 한계가 있다. 따라서 후속 연구로 시간에 대해 독립적인 심층신경망 메타모델링 기법에 대한 연구가 필요하다.

또한 본 연구에서 사용한 심층신경망 모델은 완전 연결 계층으로 구성된 가장 단순한 형태의 심층신경망 모델이기 때문에, 최근 연구되고 있는 다양한 형태의 신경망 모델을 활용하는 방안에 대해 고려할 수 있을 것으로 예상된다.

후 기

이 논문은 2019년도 미래창조과학부의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2018R1A1A1A05078730).

참고문헌 (References)

- (1) Shabana, A. A., 1997, "Flexible Multibody Dynamics: Review of Past and Recent Developments," *Multibody System Dynamics*, Vol. 1, No. 2, pp. 189~222.
- (2) Yoo, W. S., Kim, T. Y. and Jung, S., 2019, "History and Future of Multibody Dynamics," *Trans. Korean Soc. Mech. Eng. A*, Vol. 43, No. 7, pp. 483~491.
- (3) Kim, J. G., Han, J. B., Lee, H. M. and Kim, S. S., 2018, "Flexible Multibody Dynamics Using Coordinate Reduction Improved by Dynamic Correction," *Multibody System Dynamics*, Vol. 42, No. 4, pp. 411~429.
- (4) Kim, S., Choi, J., Kim, J. G., Hatakeyama, R., Kuribara, H. and Choi, J. H., 2019, "Coupled Simulation of Elasto-hydrodynamics and Multi-Flexible Body Dynamics in Piston-Lubrication System," *Advances in Mechanical Engineering*, Vol. 11, No. 12, pp. 1~13.
- (5) Han, J. B., Kim, J. G. and Kim, S. S., 2019, "An Efficient Formulation for Flexible Multibody Dynamics Using a Condensation of Deformation Coordinates," *Multibody System Dynamics*, Vol. 47, No. 3, pp. 293~316.
- (6) Han, S., Kim, J. G., Choi, J. and Choi, J. H., 2020, "Iterative Coordinate Reduction Algorithm of Flexible Multibody Dynamics Using a Posteriori Eigenvalue Error Estimation," *Applied Sciences*, Vol. 10, No. 20, Article 7143.
- (7) Sohn, J. H. and Baek, W. K., 2005, "Vehicle Dynamic Simulation Including an Artificial Neural Network Bushing Model," *Journal of Mechanical Science and Technology*, Vol. 19, No. 1, pp. 255~264.
- (8) Choi, H. S., An, J., Han, S., Kim, J. G., Jung, J. Y., Choi, J., Orzechowski, G., Mikkola, A. and Choi, J. H., 2020, "Data-driven Simulation for General-Purpose Multibody Dynamics Using Deep Neural Networks," *Multibody System Dynamics*, Vol. 51, No. 4, pp. 419~454.
- (9) Han, S., Choi, H. S., Choi, J., Choi, J. H. and Kim, J. G., 2021, "A DNN-based Data-driven Modeling Employing Coarse Sample Data for Real-Time Flexible Multibody Dynamics Simulations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 373, Article 113480.
- (10) Han, S., Choi, H. S., Choi, J., Choi, J. H. and Kim, J. G., 2020, "Study of Deep Neural Network-based Meta-Modeling Method for Real-Time Flexible Multibody Dynamics Simulation," *Proc. of the KSME Spring Conference*, pp. 77~78.
- (11) Lei, Y., Scheffer, N., Ferrer, L. and McLaren, M., 2014, "A Novel Scheme for Speaker Recognition Using a Phonetically-Aware Deep Neural Network," *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1695~1699.
- (12) Piczak, K. J., 2015, "Environmental Sound Classification with Convolutional Neural Networks," *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing*, pp. 1~6.
- (13) Acharya, U. R., Oh, S. L., Hagiwara, Y., Tan, J. H., Adam, M., Gertych, A. and San Tan, R., 2017, "A Deep Convolutional Neural Network Model to Classify Heartbeats," *Computers in Biology and Medicine*, Vol. 89, pp. 389~396.
- (14) Wang, X., Zhang, W., Wu, X., Xiao, L., Qian, Y. and Fang, Z., 2019, "Real-Time Vehicle Type Classification with Deep Convolutional Neural Networks," *Journal of Real-Time Image Processing*, Vol. 16, No. 1, pp. 5~14.
- (15) Ignatov, A., 2018, "Real-Time Human Activity Recognition from Accelerometer Data Using Convolutional Neural Networks," *Applied Soft Computing*, Vol. 62, pp. 915~922.
- (16) George, D. and Huerta, E. A., 2018, "Deep Neural Networks to Enable Real-Time Multimessenger Astrophysics," *Physical Review D*, Vol. 97, No. 4, Article 044039.
- (17) Jiang, P., Shu, L., Zhou, Q., Zhou, H., Shao, X. and Xu, J., 2015, "A Novel Sequential Exploration-Exploitation Sampling Strategy for Global Meta-Modeling," *IFAC-PapersOnLine*, Vol. 48, No. 28, pp. 532~537.
- (18) Sheikholeslami, R. and Razavi, S., 2017, "Progressive Latin Hypercube Sampling: An Efficient Approach for Robust Sampling-based Analysis of Environmental Models," *Environmental Modelling & Software*, Vol. 93, pp. 109~126.

- (19) Son, S., 2013, “Generalization of Jacobian in Quadratic Velocity Term of Modal Flexible Multibody System,” *MS Thesis*, Kyung Hee University, Republic of Korea.
- (20) FunctionBay, 2019, *RecurDyn V9R3 User Manual*, FunctionBay, Inc., Seongnam.