

로봇 매니플레이터의 동적 모델 학습을 위한 순환 신경망 기반 학습 과정

Recurrent Neural Network-based Learning Process of Dynamic Model in Robot Manipulators

신승천¹·강동엽²·백재민[†]

Seungcheon Shin¹, Dongyeop Kang², Jaemin Baek[†]

Abstract: This paper proposes a new learning process for dynamic model of robot manipulators, which is based on a recurrence neural network (RNN) algorithm with feedback structure. For a model-learning process, we first chose well-known activity functions that Sigmoid, ReLU, and Hyperbolic tangent functions were employed to confirm the suitability of the dynamic model in the RNN. Next, to make input/output variables suitable for closed-loop systems, they were set as joint angles, angular velocities, and torques for each axis of the virtual robot manipulators. Then, these variables are directly related to reference trajectories that have several changes in moving speed of end-effector while drawing circles with continuously changing radius in task space. The effectiveness of the learned dynamic model were proved by using some of collected variables. Also, it were demonstrated to verify the feasibility of the proposed process in a closed-loop robot manipulator simulation, which were compared to tracking performance in a virtual robot manipulator.

Keywords: Learned Dynamic Models, Closed-Loop Control Systems, Recurrence Neural Networks, Robot Manipulators

1. 서 론

수십 년 동안 많은 연구자는 용접^[1], 조립^[2], 수송^[3] 등 제조 산업에서의 작업 능력을 향상하고자 노력해왔고, 최근에는 군사^[4], 수중 탐사^[5], 우주 탐사^[6], 무인 탐사^[7] 등 융합 및 응용 산업에까지 높은 작업 능력을 성취하고자 많은 관심을 이끌고 있

다. 이렇게 제조 산업뿐만 아니라 여러 산업에서까지 연구 분야가 확장되는 것은 모든 산업에서의 작업 공간에서 자동화를 이루기 위함이며, 작업 능력과 효율을 크게 향상함으로써 제품에 대한 가치를 높이고 비용을 절감하는 데 그 목적을 두고 있다.

로봇 매니플레이터는 자동화를 이룩하는 데 중심이 되는 동적 시스템이고, 이동 매니플레이터^[8], 병렬 매니플레이터^[9], 케이블 구동 매니플레이터^[10], 수직 매니플레이터^[11] 등 다양한 매니플레이터가 개발되면서 많은 연구자는 로봇 매니플레이터의 원천기술을 연구하는 데 많은 노력을 기울였다. 특히, 로봇과 관련된 원천기술 중에서 작업의 정밀도와 강인성을 향상하며 그 성능을 직접 조절할 수 있는 모션(Motion) 제어 분야가 대표적이라 말할 수 있다^[12-15]. 여기서 모션을 제어하기 위해서는 로봇 매니플레이터의 내부 루프에 구현된 제어 기술이 핵심이며, 이 제어 기술은 시스템에 장착된 구동기에 직접 적용되기 때문에 제어에 대한 전문지식을 요구해 왔다.

이처럼 많은 연구자는 로봇 매니플레이터와 같은 동적 시스템에서의 구동 성능을 높이고자 제어 기법을 설계하는 데 오랜

Received : Jan. 29. 2024; Revised : Mar. 16. 2024; Accepted : Apr. 29. 2024

※ This work was supported by the National Research Foundation of Korea through the Basic Science Research Program funded by the Ministry of Education under Grant 2021R111A3059648 and was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government. [24ZD1130, Regional Industry ICT Convergence Technology Advancement and Support Project in Daegu-Gyeongbuk (Robot)].

1. Graduated Student, Department of Mechanical Engineering, Gangneung-Wonju National University, Wonju, Korea (sschun0854@gwnu.ac.kr)

2. Principal Researcher, Electronics and Telecommunications Research Institute, Daegu, Korea (kang@etri.re.kr)

† Associate Professor, Corresponding author: Department of Mechanical Engineering, Gangneung-Wonju National University, Wonju, Korea (jmbaek@gwnu.ac.kr)

시간 동안 몰두했으며, 그들 중 잘 알려진 방법으로는 선형 2차 제어기(Linear Quadratic Control)^[16], H-infinity 제어기^[17], 계산 토크 제어(Computed Torque Control)^[18] 등이 있다. 선형 2차 제어기는 시스템의 입력값과 출력값에 관한 목적 함수(Cost Function)를 두고 이를 최소화할 수 있는 이득을 구하는 데 사용된다. H-infinity 제어기는 제어 문제를 수학적으로 최적화하여 시스템의 안정도를 보장하는 최적의 이득을 설정해 주며, 시스템이 채널 간 교차적으로 다변량 결합하여 있을지라도 선형 2차 제어기와 같은 고전적인 제어 방법에 비해 시스템에 대한 강인성을 높이는 효과를 제공한다. 계산 토크 제어는 시스템에 대한 규범 모델(Nominal Model)에 근간을 두어 설정되며 규범 모델이 제공되는 시스템에서는 실무자가 적용하기에 편의성을 도모할 수 있다는 장점이 있다. 하지만, 이들은 로봇 매니플레이터와 같이 다축 연결 시스템에 적용할 때 복잡한 운동 방정식을 풀어야 한다는 어려움을 지니며, 이들은 시스템 모델에 근간을 두어 설계되어야 하므로 동적 모델에 대한 정보가 정확하지 않다면 시스템에서의 정교성과 강인성이 크게 저하되는 문제를 초래할지도 모른다.

로봇 매니플레이터는 다른 동적 시스템과 달리 마찰력 등 내부 외란에 대한 영향이 크기 때문에 제어 관련 분야의 전공자가 아닌 실무자들은 로봇 매니플레이터와 같이 복잡하고 다축 연결된 시스템의 모델을 얻기는 여간 어려운 일이 아니다. 이러한 이유로, 많은 연구자는 시스템의 정확한 모델을 습득하고자 인공지능(Artificial Intelligence) 기법에 근간을 두어 연구를 진행하기 시작하였다. 먼저, 시스템의 동적 모델을 학습하기 위해 다층 퍼셉트론 신경망(Multi-layer Perceptron Neural Network) 알고리즘^[19]을 활용한 방법이 있었다^[20]. 이 논문은 로봇 매니플레이터의 동적 모델을 직접 학습하기보다 역기구학(Inverse Kinematics)/순기구학(Forward Kinematics)에 대해 동적 모델을 간접적으로 학습하는 방법을 제시하였으며, 자코비안 행렬(Jacobian Matrix)을 학습하는 목적으로 사용되었다. 또한, 로봇 매니플레이터의 동적 모델을 알지 못하더라도 토크 데이터와 같은 상태변수로부터 이상 상태를 감지할 수 있도록 다층 퍼셉트론 신경망 알고리즘이 적용되었다^[21]. 이 방법은 복잡한 동적 모델을 학습하지 않더라도 특이점(Singularity)과 같은 위태로운 상태를 피하는 것을 돕는 장점을 제공하였다. 또 다른 연구 방향으로는 로봇 매니플레이터의 동적 모델을 학습하고자 다층 퍼셉트론 신경망 알고리즘을 이용하였다^[22]. 이 논문은 로봇 매니플레이터가 지닌 질량 관성 모멘트(Moment of Inertia), 코리올리 힘(Coriolis Force), 원심력(Centrifugal Force), 중력(Gravity) 그리고 마찰력(Friction)과 관련된 수식들을 별도로 학습하여 각각 학습된 모델을 시뮬레이터에 적용할 수 있다는 이점을 가진다. 하지만, 역기구학/순기구학과 이상 상태 감지를 목적으로 적용된 다층 퍼셉트론 신경망 알고리즘은 기존 학

습된 입력값에 근간을 둔 출력값을 제공하기 때문에 긍정적인 결과를 도출할 여지가 충분하지만, 시간과 환경에 따라 급변할 수 있는 상태변수들을 지닌 로봇 매니플레이터의 동적 모델을 직접 학습하는 과정에서 동적 모델에 대한 학습 정확도를 높이는 쉬운 일이 아닐지도 모른다. 다시 말해, 로봇 매니플레이터는 각도, 각속도 그리고 각가속도에 기반을 둔 동적 모델을 지니기에 각도의 변화량과 각가속도의 변화량이라는 에너지의 변화에 동반할 수 있는 학습 방법이 요구될 수밖에 없다.

앞서 언급된 단점을 줄이고 로봇 매니플레이터의 동적 모델을 학습하는 데 있어서 더욱 긍정적인 결과를 도출하고자, 순환 신경망(Recurrent Neural Network) 알고리즘^[23]에 근간을 둔 학습 과정이 제시되었다^[24]. 이 연구는 오일러-랑그랑주 방정식을 통해 로봇 매니플레이터를 모델링했고, 이 수식에 근간을 둔 시뮬레이션을 통해 순환 신경망을 이용하여 로봇 매니플레이터의 동적 모델을 학습하였다. 순환 신경망은 학습을 거듭할수록 오래전의 정보에 대한 영향이 퇴색되며, 현재 이산 시간으로 직전 상태에 대한 정보를 전달할 수 있으므로 동적 모델이 지닌 에너지에 대한 모델을 학습하는 데 유리함은 이 연구를 통해 입증되었다. 하지만, 이 논문에서 제시한 학습 과정은 로봇 매니플레이터가 지닌 잡음과 같은 내부 외란에 대한 요소들을 고려하지 않았을 뿐만 아니라 시스템에서의 정보를 활용하지 않았기에 로봇 매니플레이터에 이 학습 과정을 적용한다면 학습 효과가 저하될 여지가 있으며 실용성에 대한 문제까지 초래할지도 모른다.

본 논문에서는 순환 신경망 알고리즘을 활용하여 로봇 매니플레이터의 동적 모델을 학습하는 과정을 제안한다. 본 연구는 로봇 매니플레이터의 동적 모델을 학습하기 위한 방향성을 제시하며, 학습된 동적 모델에 근간을 둔 피드백 제어 시뮬레이션을 제공할 수 있도록 하는 데 목적을 두고 있다. 먼저, 로봇 매니플레이터의 동적 모델을 학습하는 데 적합한 신경망 구조를 확인하고자 피드 포워드(Feed-forward) 구조를 가진 다층 퍼셉트론 신경망 알고리즘과 피드백 구조를 가진 순환 신경망 알고리즘을 선정하였고, 이들에 근간을 두어 동적 모델에 대한 학습 성능을 비교하였다. 두 번째로는 로봇 매니플레이터의 동적 모델을 학습하면서 신경망 알고리즘을 통한 동적 모델의 연속성 및 신경망의 표현성을 높일 수 있는 시그모이드 함수(Sigmoid Function), 렐루 함수(ReLU Function) 그리고 하이퍼볼릭 탄젠트 함수(Hyperbolic Tangent Function)에 근간을 두어 적합한 비선형 활성화 함수를 선정하였다. 세 번째로 본 연구에서는 신경망 알고리즘을 이용한 학습 과정과 피드백 제어 시뮬레이션에 적합한 입-출력 변수를 선택하였고, 로봇 매니플레이터의 축별 각도, 각속도 그리고 토크 변수를 적용하였다. 이때, 각가속도 변수는 신호의 잡음 등 외부 외란에 대한 요소를 증폭시킬 여지가 크며^[12], 유효하지 않은 출력값으로 인해 활성화 함수에

서 학습 효과가 저하될 수 있기에 본 연구에서는 각가속도 변수를 학습 과정에 반영하지 않았다. 네 번째로는 로봇 매니플레이터의 동적 모델의 학습 성능을 향상하게 시킬 수 있는 축별 각도, 각속도 그리고 토크 변수를 얻고자, 카테시안 공간에서 x축, y축 그리고 z축으로 움직임에 대한 속도 변화를 주면서 원하는 목표 궤적을 설정하였다. 다시 말해, 로봇 매니플레이터의 움직임에 대한 반경과 속도 변화는 로봇 매니플레이터의 관성 모멘트와 직접적으로 관련되어 있고, 관성 모멘트의 크기는 변칙적인 토크에 대한 정보를 제공할 수 있기에 범용적인 변수 데이터를 수집하는 데 용이하다. 마지막으로로는 로봇 매니플레이터의 학습된 동적 모델에 대한 성능을 비교 및 분석하였다. 이때, 학습된 동적 모델은 수집된 변수 정보의 일부를 검증용 데이터로 분류하여 학습된 모델에 대한 성능을 증명하였다. 게다가, 학습된 동적 모델에 근간을 둔 피드백 제어 시뮬레이션을 진행하여 제안된 학습 과정으로부터 얻어진 동적 모델에 대한 성능을 추가로 입증하였다. 이때, 실무에서 많이 사용되는 비례-미분(Proportional-derivative) 제어를 구현하였으며, 가상의 로봇 매니플레이터에도 동일한 비례-미분 제어를 적용하여 학습된 동적 모델을 활용할 시뮬레이션의 결과와 유사성을 분석하였다.

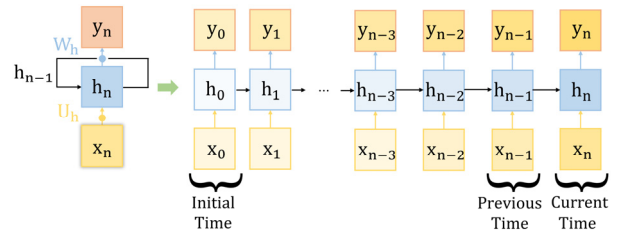
본 논문의 나머지 부분은 다음과 같이 구성된다. 2장에서는 일반적인 순환 신경망과 활성화 함수 그리고 제안된 동적 모델 학습 과정을 소개하고, 3장에서는 제안된 학습 과정을 적용하여 순환 신경망 알고리즘에 근간을 둔 동적 모델에 대한 학습 결과를 다층 퍼셉트론 신경망 알고리즘의 동적 모델 학습 결과와 비교한다. 4장에서는 활성화 함수와 노드 및 은닉층 수에 따른 추가적인 실험을 진행하며, 5장에서는 본 논문의 결과를 정리하며 마무리한다.

2. 제안된 신경망 기반 학습 과정

2.1 신경망 알고리즘 선정

순환 신경망은 시계열로 구성된 데이터를 학습하고 데이터의 속성을 예측하는데 매우 높은 효과를 도출한다고 알려진 대표적인 인공 신경망 중 하나이다^[23]. 순환 신경망은 입력층과 은닉층 그리고 출력층의 3단계로 이루어지며, 각 층은 서로 완전 연결 계층(Fully connected layer)의 구조로 구성되어 있다.

[Fig. 1]은 순환 신경망을 도식화하여 입출력을 나타낸 것으로서 이산 시간의 변화에 따른 입출력 관계에 근간을 두어 직전 상태에 대한 데이터 정보를 활용하여 현재 상태에 대한 데이터 정보를 예측하기 위해 전달하는 과정을 거친다. 이때, 이 신경망은 순환 과정을 거듭할수록 이전 시간에 대한 정보의 가중치



[Fig. 1] Model structure of recurrent neural network

(Weight Factor)가 줄어드는 성질을 보이며, 많은 문헌에서는 이러한 현상을 장기 의존(Long-term Dependency) 현상이라 명명한다^[25]. 장기 의존 현상에 대해서는 [Fig. 1]에서 화살표와 배경색의 짙음을 통해 가시적으로 표현하였으며, 이와 같은 순환 신경망의 과정을 수식화하면 다음과 같다:

$$h_n = f(U_h x_n + W_h h_{n-1}) \tag{1}$$

$$y_n = O_h h_n \tag{2}$$

식 (1)에서 $n \in \mathbf{R}$ 은 이산 시간에 대한 횟수를 나타낸다. $x_n \in \mathbf{R}^m$ 과 $h_n \in \mathbf{R}^L$ 는 각각 순환 신경망의 입력과 은닉 상태를 의미하며, 작동점($n=0$)에서의 값은 영(Zero)으로 설정된다. $U_h \in \mathbf{R}^{L \times m}$, $W_h \in \mathbf{R}^{L \times L}$, $O_h \in \mathbf{R}^L$ 는 모두 순환 신경망의 파라미터를 나타내고 있으며, y_n 는 순환 신경망 알고리즘의 출력값을 표현하고 있다. 여기서, \cdot_n 과 \cdot_{n-1} 은 각각 n 번째와 $n-1$ 번째 이산 시간에서의 정보를 나타내며, m 은 입력 데이터와 은닉 상태의 차원을 의미한다. 위 수식에서 확인할 수 있듯이 순환 신경망은 직전 이산 시간에서의 정보를 받아 현재 이산 시간으로 정보를 전달하며 시간에 대해 순차적인 데이터의 처리가 이루어지게 된다. 이는 물리적으로 정보의 변화량을 의미하기에 순차적인 데이터의 처리 과정을 수학적으로 표현한다면 미분 방정식에 근간을 둘 수 있다. 이때, 미분 방정식은 시스템에 대한 동적 모델을 나타냄으로써 순환 신경망의 시스템 모델을 일반적으로 잘 알려진 로봇 매니플레이터의 동적 모델과 유사한 형태로 해석할 수 있으며 그 수식은 아래와 같다^[12]:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(q, \dot{q}) = \tau + \tau_d \tag{3}$$

식 (3)에서 $q \in \mathbf{R}^p$, $\dot{q} \in \mathbf{R}^p$ 그리고 $\ddot{q} \in \mathbf{R}^p$ 는 각각 로봇 매니플레이터의 관절의 각도, 각속도 그리고 각가속도를 나타내며, $M(q) \in \mathbf{R}^{p \times p}$ 는 관성 모멘트, $C(q, \dot{q}) \in \mathbf{R}^p$ 는 코리올리 힘과 원심력, $G(q) \in \mathbf{R}^p$ 는 로봇 매니플레이터에 작용하는 힘, $F(q, \dot{q}) \in \mathbf{R}^p$ 는 마찰력, $\tau \in \mathbf{R}^p$ 와 $\tau_d \in \mathbf{R}^p$ 는 각각 로봇 매니플레이터의 관절에 인가되는 제어 입력과 외부 외란을 의미한다.

이때, P 는 로봇 매니플레이터의 관절 번호를 나타내고 있다.

앞서 언급된 $\dot{\mathbf{q}}$ 은 \mathbf{q} 의 변화량이라는 물리량을 나타내며, $\ddot{\mathbf{q}}$ 도 앞서 언급된 \mathbf{q} 와 마찬가지로 $\dot{\mathbf{q}}$ 의 변화량이라는 물리적인 속성을 지닌다. 다시 말해, $\dot{\mathbf{q}}$ 과 $\ddot{\mathbf{q}}$ 의 변화량이라는 물리적 성질은 각각 \mathbf{q} 와 $\dot{\mathbf{q}}$ 가 변화할 수 있는 에너지를 내포하며, 내재적으로 지닌 에너지는 동적 움직임을 보여주는 역할을 한다. 또한, 시스템에서의 동적 움직임은 시간에 따라 순차적으로 변화하기에 시간에 종속적인 상태변수의 경우에는 순환 신경망의 형태와 비슷한 움직임을 가질 수밖에 없다. 이러한 이유로, 로봇 매니플레이터의 동적 모델은 순환 신경망에 근간을 두어 수집된 시계열 데이터를 활용한 학습이 진행된다면 다른 학습 기법들보다 뛰어난 성능을 발휘할 여지가 충분히 있다²⁶⁾.

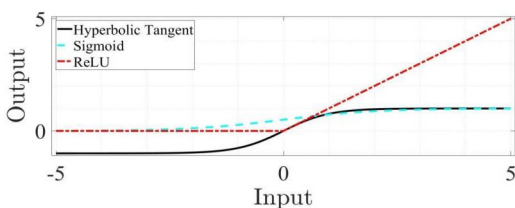
2.2 활성화 함수 선정

활성화 함수에는 하이퍼볼릭 탄젠트(Hyperbolic Tangent) 함수와 시그모이드(Sigmoid) 함수 그리고 렐루(ReLU) 함수가 대표적으로 사용되며, 함수들의 수식은 [Table 1]에서 확인할 수 있다.

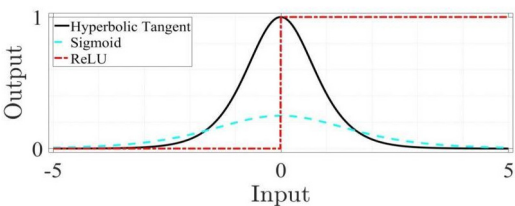
[Fig. 2]를 통해 알 수 있듯이 하이퍼볼릭 탄젠트 함수는 -1부터 1 사이의 값을 출력하는 반면에 시그모이드 함수는 0부터

[Table 1] Activation functions and their equations

Activation Function	Equation
Hyperbolic Tangent	$T(\eta) = \frac{e^\eta - e^{-\eta}}{e^\eta + e^{-\eta}}$
Sigmoid	$S(\eta) = \frac{1}{1 + e^{-\eta}}$
ReLU	$R(\eta) = \max(0, \eta)$



[Fig. 2] Output value of activation functions



[Fig. 3] The derivative of activation functions

1 사이의 값을 출력하며, 렐루 함수는 영(Zero)보다 큰 값이 입력되었을 때만 의미 있는 출력값을 도출하는 것을 알 수 있다. 여기서, 입력값과 출력값은 원시 데이터의 크기가 아닌 데이터에 대한 크기를 정규화하였기에 입력값과 출력값의 단위는 사용자의 설정에 따라서 달라질 수 있다. 또한, [Fig. 3]에서 볼 수 있듯이, 입력값이 영(Zero)인 부근에서 하이퍼볼릭 탄젠트 함수의 미분값이 시그모이드 함수의 미분값에 비해 상대적으로 큰 값을 도출한다. 다시 말해, 입력이 영(Zero)인 부근에서 하이퍼볼릭 탄젠트 함수는 시그모이드 함수에 비하여 출력값의 범위가 더 클 뿐만 아니라 미분값 변화량도 더 큰 값을 나타내기 때문에 하이퍼볼릭 탄젠트 함수는 순환 신경망의 학습에서 기울기 소실 문제를 예방하는 데 도움을 줄 수 있다. 게다가, [Fig. 2]에서 확인할 수 있듯이, 렐루 함수는 입력값과 출력값이 서로 선형성을 가지고 있으므로 이 함수의 미분값은 영(Zero)점을 기준으로 계단 함수(Step Function) 형태를 보여주고 있다. 이와 같은 이유로, 렐루 함수는 영(Zero)보다 작은 값에서 기울기가 소실되는 문제가 발생할 여지가 있다.

2.3 데이터 속성 및 동적 모델 학습 과정

본 절에서는 순환 신경망에 근간을 두어 로봇 매니플레이터의 동적 모델을 학습하기 위한 제안된 학습 과정을 소개하고자 한다. 제안된 학습 과정에서는 [Fig. 1]에서 보여준 일반적인 순환 신경망이 사용되었으며, 각 축에서 얻은 각도 $\mathbf{q}_{n-1} \in \mathbf{R}^m$, 각 속도 $\dot{\mathbf{q}}_{n-1} \in \mathbf{R}^m$, 토크 $\boldsymbol{\tau}_{n-1} \in \mathbf{R}^m$ 가 입력 데이터로 사용됨과 동시에 출력 데이터로는 하나의 축의 각도 $\mathbf{q}_{n-1} \in \mathbf{R}$ 가 사용되었다. 다시 말해, 로봇 매니플레이터의 동적 모델을 학습하는 과정에서 $\mathbf{q}_i \in \mathbf{R}^m$ 처럼 다출력(Multi-output) 데이터를 이용한 것이 아닌 개별 출력(Single-output) 데이터를 사용하였다. 이처럼 설계된 이유는 학습된 동적 모델에 대한 폐루프 제어 시뮬레이션(Closed-loop Control Simulation)을 구성할 때 로봇 매니플레이터의 각 관절에 장착된 각도 센서에서 얻는 데이터의 취득 과정과 동일하게 구성하기 위함이다. 또한, 제안된 학습 과정에서 활성화 함수는 앞 절에서 소개한 순환 신경망 알고리즘의 기울기 소실 문제를 예방하고자 하이퍼볼릭 탄젠트 함수를 채택하였으며, 이 함수의 적절성에 대한 평가는 다음 장에서 시그모이드 함수와 렐루 함수가 사용되었을 때의 학습 성과와 비교 및 분석을 통해 진행되었다. 게다가, 제안된 학습 과정은 학습 회차마다 입력층에서 출력층으로 정보를 전달하면서 가중치를 갱신하는 순전파 과정을 거치며, 목표값과 순환 신경망 알고리즘 출력값의 오차를 토대로 출력층에서 시작하여 입력층으로 정보를 전달하면서 다시 한번 가중치를 갱신하는 역전파 과정을 거치도록 설계되었다. 또한, 과적합(Over fitting)으로 인해

[Table 2] Training procedure of recurrent neural network model employed in proposed learning method

Input: Normalized data set $\{\mathbf{q}_{n-1}, \dot{\mathbf{q}}_{n-1}, \boldsymbol{\tau}_{n-1}\}$
Output: Normalized data $q_{:,n}$
Model: $(\mathbf{U}_h, \mathbf{W}_h, \mathbf{O}_h)$
Initialize hidden state $\mathbf{h}_n = 0$
Initialize Count Count = 0
1: for $n = 1$ to N do
2: $\mathbf{h}_n = f(\mathbf{U}_h \mathbf{x}_n + \mathbf{W}_h \mathbf{h}_{n-1})$
3: $\hat{y}_n = \mathbf{O}_h \mathbf{h}_n$
4: $L_n \leftarrow \text{MSE} = \frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2$
5: if $L_n \geq L_{n-1}$ then
6: Count = Count + 1
7: end if
8: if Count ≥ 3 then
9: break;
10: end if
11: end for

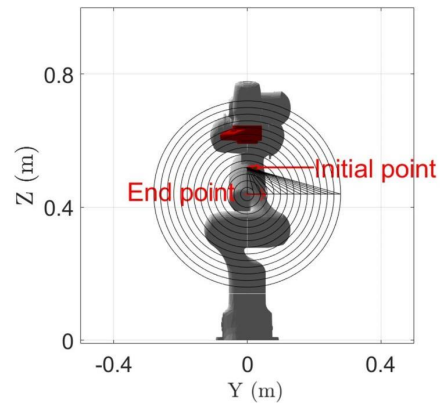
학습된 순환 신경망 모델의 일반화 오차가 증가하는 것을 예방하고자 손실 함수의 출력값이 3회 이상 줄어들지 않으면 미리 설정된 학습 횟수를 무시하고 조기에 학습이 종료되도록 모델을 구성하였다. 이러한 순환 신경망 알고리즘의 학습 과정은 [Table 2]와 같이 요약될 수 있다.

독자의 이해를 돕기 위해, [Table 2]의 학습 과정은 은닉층이 한 개만 존재할 때의 학습 과정을 나타내었다. 여기서, f 는 활성화 함수인 하이퍼볼릭 탄젠트 함수이며, L_n 은 손실 함수의 출력값을 의미한다. 본 논문에서 제안된 방법의 실험에 관한 상세 설명 및 결과는 다음 장에서 확인할 수 있다.

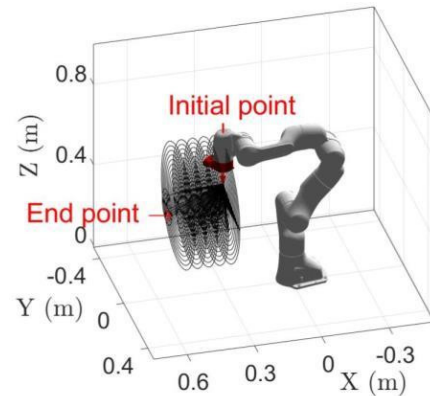
2.4 데이터 수집 방법

본 연구에서 사용된 시스템은 7축 다관절 로봇 매니플레이터 기반 가상 시스템으로써, 정현파(Sinusoidal Wave) 함수인 사인(Sine)과 코사인(Cosine) 함수를 사용하여 y-z 좌표 평면 위에서 다양한 크기의 원형 목표 경로([Fig. 4])를 설정하였다. 이러한 설정 방법은 실제 산업에서 사용되는 로봇 매니플레이터의 작업 범위(Task Space)가 구(Sphere) 안에서 이뤄지는 영역을 모사하기 위하여 제시되었고, 이를 기반으로 각 축에서 측정되는 각도와 각속도 그리고 토크에 대한 상태변수 데이터를 수집하여 동적 모델에 대한 학습을 진행하였다. 본 연구에서 사용된 로봇 매니플레이터는 1 kHz의 제어 주기에 근간을 두어 실험을 진행하였고, 상태변수 데이터에 대한 정보를 취득하는 대역폭도 제어 주기와 동일하게 설정되었다.

[Fig. 4]에서 도식화된 목표 궤적에 대해 세부적으로 설명하



[Fig. 4] Desired trajectories in end-effector of robot manipulator on y-z axis



[Fig. 5] Desired trajectories of robot manipulator in a task space

자면, 로봇 매니플레이터의 y-z 좌표 내 목표 경로를 추종한 로봇 매니플레이터의 엔드이펙터는 초기점(Initial Point)에서부터 반지름이 커지도록 목표 궤적을 구성하였고, 이와 같은 과정을 x축으로 0.05 (m)를 이동하면서 y-z축 좌표 내 목표 경로를 추종하는 움직임을 반복하도록 설정하였다. 그 후에 더 이상 경로를 진행하기 어려운 위치(기구학적 특이점)에 도달하였을 때 동작을 정지하도록 설정하였다. 이처럼 설정된 로봇 매니플레이터의 엔드이펙터의 작업 공간 내 주어진 전체 경로는 [Fig. 5]에서 확인할 수 있다.

로봇 매니플레이터의 엔드이펙터가 앞서 언급된 목표 경로를 추종하는 속도는 2 m/s, 3 m/s, 4 m/s로 설정하여 로봇 매니플레이터의 상태변수 데이터를 수집하였다. 로봇 매니플레이터의 엔드이펙터의 속도 변화는 로봇 매니플레이터의 질량 관성 모멘트와 코리올리 힘 그리고 원심력의 변화량에도 직접적으로 영향을 미치기 때문에 엔드이펙터의 속도 변화까지 고려한 상태변수 데이터를 취득하였다. 이처럼, 주어진 경로는 자코비안 함수를 이용하여 관절 공간의 목표값으로 변환되며, 관절 공간에서의 비례-미분 제어기의 적용을 통해 로봇 매니플

[Table 3] Initial position and joint rotation range by joints in robot manipulator

Joint No.	Initial Position (Deg)	Joint Rotation Range (Deg)
1	0	-19.5601 ~ 37.3917
2	-45	-48.8001 ~ 12.4098
3	0	-35.1452 ~ 27.4282
4	-135	-172.9359 ~ -52.6766
5	0	-20.4453 ~ 14.0896
6	90	59.8013 ~ 155.8909
7	0	-30.6484 ~ 39.6177

레이터가 계획된 경로를 추종할 수 있도록 구성하였다. 로봇 매니퓰레이터가 계획된 경로를 추종할 때, 축별 초기 위치와 회전 각도 범위는 [Table 3]과 같다.

본 실험에서 동적 모델을 학습하기 위해 수집된 데이터는 52개의 데이터 셋(Data Set)으로 설정되었으며, 6,959,000개의 표본(Sample)을 수집하여 본 논문에서 소개한 로봇 매니퓰레이터의 동적 모델을 학습하는 데 사용되었다. 이 데이터 셋 중 하나는 목표 경로를 추종하면서 앞서 설정된 추종 속도들과 다른 임의의 속도를 설정하여 기존 데이터 셋에 없는 정보를 가진 상태변수 데이터를 수집하였다. 이 데이터 셋을 취득한 목적은 본 논문에서 적용된 순환 신경망 모델이 특정 데이터에 편중되면서 저하되는 학습 성능을 완화하기 위함이며, 변칙적인 데이터 셋의 다양성은 학습 성능을 높여주는 촉진제 역할을 할 여지가 충분히 있다.

2.5 하이퍼 파라미터 선정

하이퍼 파라미터의 선정 기준으로는 평균 절대 오차(Mean-absolute Error, MAE)와 제곱 평균 제곱근 오차(Root-mean-square Error, RMSE) 그리고 평균 절대 비율 오차(Mean Absolute Percentage Error, MAPE)의 크기를 기준으로 하였다. 평균 절대 오차는 모든 절대 오차의 평균으로써 오차에 비례하는 수치를 가지며, 결괏값이 작을수록 더 나은 성능을 제공함을 의미한다. 제곱 평균 제곱근 오차는 이상점이 존재할 때, 이에 대한 영향을 줄일 수 있는 측정 지표이며, 이 역시 결괏값이 작을수록 더 나은 성능을 제공한다. 또한, 평균 절대 비율 오차는 예측 정확도를 비율로 나타낼 수 있는 측정 지표로써, 예측 오차에 대해 비율로 제시할 수 있고 다양한 모델과 데이터에 대한 성능 비교에 적합함이 입증되어왔다^[27]. 이 측정 지표들에 대한 수식은 [Table 4]에서 확인할 수 있다.

위 수식들에서 y_j 는 목표값이며, \hat{y}_j 는 순환 신경망 알고리즘의 출력값을 나타낸다. 여기서, 평균 절대 오차와 제곱 평균 제

[Table 4] Measurement Metrics

Content	Equation
MAE	$\frac{1}{n} \sum_{j=1}^n y_j - \hat{y}_j $
RMSE	$\sqrt{\frac{\sum_{j=1}^n (y_j - \hat{y}_j)^2}{n}}$
MAPE	$\frac{100}{n} \sum_{j=1}^n \left \frac{y_j - \hat{y}_j}{y_j} \right $

[Table 5] General setup method of hyper-parameters based on recurrent neural network in robot manipulators

Content	Setting Method
Optimizer	SGD, RMSProp, Adam, etc.
Data-bandwidth	1 ~ 1000 Hz
Activation Function	Equations in [Table 1]
Loss Function	Equations in [Table 4]
The Number of Hidden Layers	1 ~ 20
The Number of Nodes	A form of 2^n where $n > 0$ is a constant
Batch Size	A form of 2^m where $m > 0$ is a constant
Epoch	50 ~ 200

곱근 오차는 학습 과정에서 손실 함수와 측정 지표로써 활용되며, 평균 절대 비율 오차는 학습이 완료된 이후 검증 데이터를 사용하여 학습 모델을 확인하기 위해 적용되었다. 이와 같은 이유로, 본 논문에서는 평균 절대 오차와 제곱 평균 제곱근 오차에 대해 먼저 비교한 다음 순차적으로 평균 절대 비율 오차에 대해 비교한다.

순환 신경망 알고리즘을 활용하여 로봇 매니퓰레이터의 동적 모델을 학습하는 과정에 최적의 성능을 도출하기 위한 파라미터를 설정하는 방법은 [Table 5]에 명시되었다.

[Table 5]에서 소개된 항목들을 설정하는 방법은 학습 시간을 감소시키고, 학습 데이터에 내재된 잡음을 줄이며, 과적합을 예방하는 데 도움을 준다고 잘 알려져 왔다^[28]. 특히, 노드 수와 배치 크기의 경우, 2의 거듭제곱 형태로 설정하는 것은 학습 과정에서 메모리를 할당하는 것이 매우 효과적이며 빠른 연산 처리가 가능함이 증명됐다^[29,30]. 또한, 데이터에 대한 대역폭 크기, 은닉층 수 그리고 학습 횟수는 그 수가 커질수록 모델을 학습하는 과정에서 컴퓨팅 메모리를 할당하는 것이 매우 취약해짐과 동시에 그들의 수와 학습 성능이 비례적인 관계가 아니기에 수많은 연구자는 시행착오를 거쳐 이들을 설정해왔다^[31,32]. 덧붙여, 수식 (3)에서 소개된 로봇 매니퓰레이터의 동적 모델은 다른 시스템들과 달리 모든 관절이 서로 커플링(Coupling) 되어있을 뿐만 아니라 임의의 한 관절에서의 영향이 모든 관절에

[Table 6] Hyper-parameters of training procedure

Content	Setting Value
Optimizer	Adam
Data-bandwidth	10 Hz
Activation Function	Hyperbolic Tangent
Loss Function	MAE
The Number of Hidden Layers	2
The Number of Nodes	128 → 64 → 32 → 1
Batch Size	32
Epoch	100

전파되도록 구성되어 있으므로 로봇 매니퓰레이터의 질량 관성 모멘트에 대한 작은 변화에도 모든 관절에 큰 악영향을 줄 수 있다^[33]. 결과적으로 로봇 매니퓰레이터의 동적 모델에 적합한 하이퍼 파라미터의 설정값을 찾는 것은 불가피하기에, 본 연구에서는 많은 문헌에서 사용된 하이퍼 파라미터 설정 방법에 근간을 두어 학습을 진행하였으며, [Table 5]에서처럼 로봇 매니퓰레이터의 동적 모델에 대한 학습 성능을 높일 수 있는 설정 방법을 제시하였다.

본 논문에서는 [Table 5]의 설정 범위에 근간을 두어 로봇 매니퓰레이터의 동적 모델을 학습하기 위한 최적의 하이퍼 파라미터를 채택하였고, 로봇 매니퓰레이터의 1축 관절부터 7축 관절까지 모든 관절에 대해 동일한 하이퍼 파라미터를 적용하였다. 이때, 본 논문에서 선택된 하이퍼 파라미터에 대한 최적성을 입증하기 위해 학습 과정에서 가장 영향을 많이 주는 것으로 잘 알려진 노드 수와 은닉층 수 그리고 활성화 함수에 따른 학습된 성능을 4장에서 토론할 것이며, 설정된 하이퍼 파라미터 들은 [Table 6]에서 확인할 수 있다.

이때, 로봇 매니퓰레이터가 지닌 신호의 경우에는 화학적인 신호와 같이 상대적으로 급진적이고 변칙성이 심하지 않기 때문에 10 ~ 100 Hz의 대역폭을 지닌 데이터에서는 하이퍼 파라미터 설정값에 대한 차이가 크게 보이지 않으며 데이터의 대역폭이 커질수록 오히려 최적의 하이퍼 파라미터를 찾아내는 학습 시간이 기하급수적으로 상승하기에 데이터의 대역폭이 클수록 실용성이 저하되는 문제와 직결될 여지가 충분히 있다. 이와 같은 이유로, 본 논문에서 학습의 경향을 쉽게 찾고 작업의 효율성을 높일 수 있는 하이퍼 파라미터를 찾고자, 사용된 데이터의 대역폭은 10 Hz로 축소(Down-sampling)하여 진행하였다.

3. 실험

3.1 실험 설명

본 학습 과정에서는 최적의 하이퍼 파라미터를 찾기 위해 사

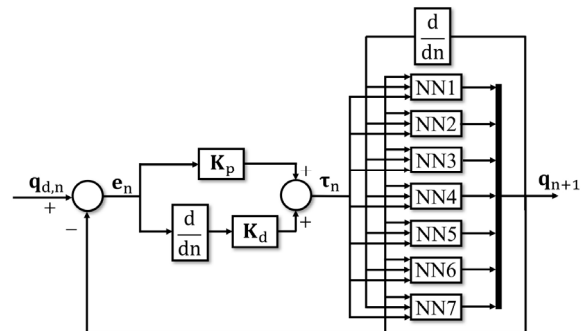
용된 데이터의 대역폭을 10 Hz로 설정하였으나, 로봇 매니퓰레이터의 동적 모델을 학습하는 데 있어서 데이터의 대역폭을 축소하는 것은 시스템이 지닌 모델 정보를 손실시킬 여지가 있다. 이러한 문제를 완화하기 위해, 로봇 매니퓰레이터의 동적 모델을 학습할 때는 데이터의 대역폭을 100 Hz로 설정하여 진행하였다.

본 논문에서 적용된 순환 신경망 모델의 입력은 총 21개의 특성(Feature)을 가지도록 구성하였으며, 입력마다 축별 각도, 각속도 그리고 토크 데이터의 1개 샘플을 적용하였다. 이때, 출력 데이터에는 n 번째 이산 시간에서 개별 축에 대한 각도 데이터를 적용하였으며, 1 축부터 7 축까지 순차적으로 로봇 매니퓰레이터의 동적 모델을 학습하는 과정을 거쳤다.

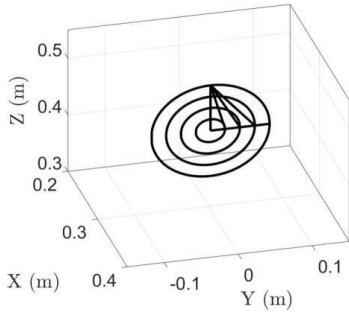
학습된 동적 모델의 성능을 공정하게 분석하기 위해, 동적 모델을 학습하는 데 가장 많이 사용되어 온 다층 퍼셉트론 신경망(Multi-layer Perceptron Neural Network, MLPNN) 알고리즘^[19]을 응용하였고, 비교군도 마찬가지로 본 논문에서 제안된 방법과 동일한 과정을 통하여 하이퍼 파라미터를 선정하였다. 그리고 두 알고리즘에서 학습된 동적 모델의 정확도는 다음과 같이 두 가지 과정을 통해 판별하였다:

- ① 목פות값과의 오차 데이터 비교 및 분석.
- ② 학습된 동적 모델에 근간을 둔 페루프 제어 시뮬레이터 (Software-in-the-loop system, SILS) 구성.

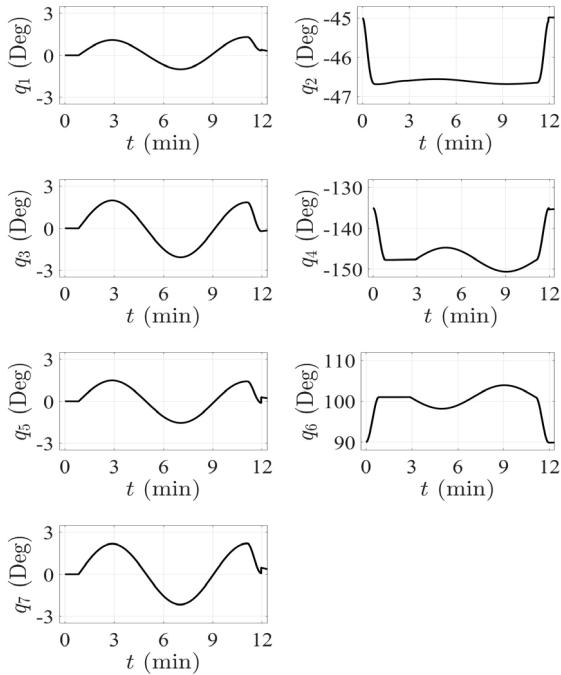
②의 경우에는 ①의 경우와 달리 실험 결과와 직접적으로 비교할 수 있다는 장점을 가지고 있다. ②에서 사용된 방법은 객관적으로 학습된 동적 모델에 대한 가용성과 실용성을 보일 수 있는 지표로서, 학습된 순환 신경망 알고리즘 기반 동적 모델과 학습된 다층 퍼셉트론 신경망 알고리즘 기반 동적 모델을 전달 함수로 활용하여 비례-미분 제어를 페루프로 구성하였다 ([Fig. 6]). 이때, 비교군과의 공정한 비교 및 검토를 위해 비례-미분 제어기의 이득 값과 제어 주기는 서로 동일하게 설정하였



[Fig. 6] The schematic diagram of dynamics model learning-based closed-loop system



[Fig. 7] The desired position-tracking trajectories based on circle-position in task space



[Fig. 8] The desired angles based on circle-position in joint space

으며, 설정된 이득 값과 제어 주기는 가상의 로봇 매니퓰레이터에 적용된 설정값과 1 kHz로 동일하게 설정하였다.

[Fig. 6]에서 $q_{i,n}$, $q_{i,n+1}$ 그리고 τ_n 은 각각 n 번째 목표 각도와 n+1 번째의 예측 각도, n 번째 입력 토크를 의미한다. e_n 은 목표 각도와 예측 각도의 오차를 나타내며, K_p 와 K_d 는 각각 비례 이득 값과 미분 이득 값을 일컫는다. NN1부터 NN7은 로봇 매니퓰레이터의 1 축부터 7 축까지 학습된 동적 모델을 나타내고 있다.

[Fig. 7]은 작업 공간에서 로봇 매니퓰레이터의 엔드이펙터에 대한 위치 움직임을 보이며, 시작 좌표(0.3 m, 0 m, 0.5 m)에서부터 y-z축 평면에 반지름을 3 cm씩 점차 증가시키면서 원을 그리는 위치 경로를 생성하였고, 로봇 매니퓰레이터의 동적 모델을 학습하기 위해 이 과정에서의 모든 좌표에 대한 데이터를 추출하였다. 또한, 추출된 위치 데이터는 자코비안 함수를 이용하여 로봇 매니퓰레이터의 각 관절에 대한 목표 각도

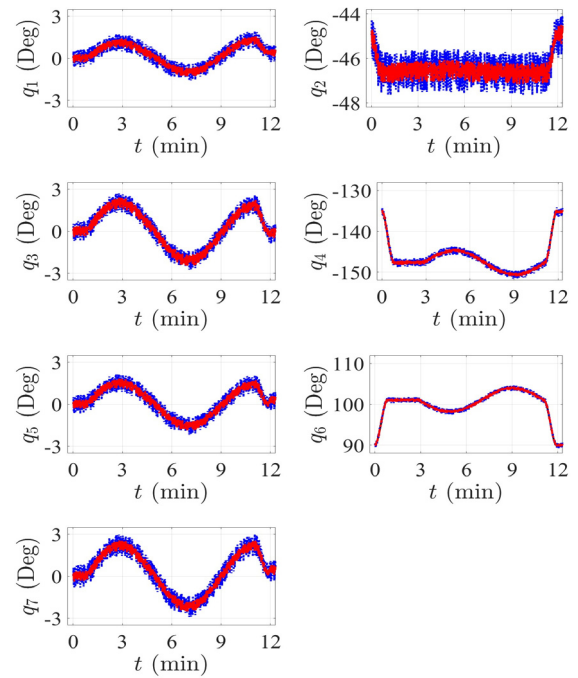
([Fig. 8])로 변환되었으며, 변환된 데이터에 근간을 두어 데이터 전처리 작업을 수행했다.

3.2 실험 결과

[Fig. 9]는 두 신경망 알고리즘에 학습된 동적 모델의 목표값에 출력값을 덧붙여 표현하고 있다. [Fig. 9]에서 제공된 가시적인 결과를 토대로 본문에서 제시된 시간에 종속적인 상태변수와 유사한 움직임을 나타내는 순환 신경망 알고리즘이 다층 퍼셉트론 신경망 알고리즘에 비해 로봇 매니퓰레이터의 동적 모델을 학습하는 데 더 효과적임을 확인할 수 있다.

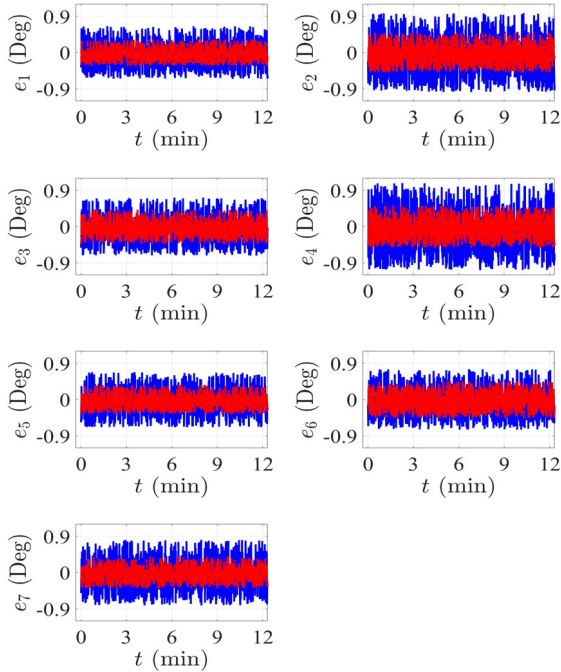
[Fig. 10]은 두 신경망 알고리즘에 학습된 동적 모델의 목표값과의 학습 오차를 나타내며, [Fig. 9]에서 보여준 값들의 정확도를 판별하는 지표가 된다. [Fig. 10]에서 볼 수 있듯이, 1 축, 3 축, 5 축, 7 축의 학습 오차에 비해 2 축, 4 축, 6 축의 출력 오차가 상대적으로 크지만, 모든 축에 대해 1 (Deg) 이내의 작은 학습 오차를 가진다. 이때, MLPNN 기반 학습 모델과 RNN 기반 학습 모델에 대한 정확도는 각각 [Table 7]과 [Table 8]에서 확인할 수 있다.

[Table 7]과 [Table 8]에서 학습 오차를 보면, 두 알고리즘 기반 학습 결과는 모든 축끼리의 유사한 경향을 보이고 있으며, 평균 절대 오차와 제곱 평균 제곱근 오차에서 그 유사성을 확인할 수 있다. 그리고 다층 퍼셉트론 신경망 알고리즘에 학습된 동적 모델의 오차보다 순환 신경망 알고리즘에 학습된 동



[Fig. 9] Comparison of model-learning results based on RNN algorithm (red dotted line) and MLPNN algorithm (blue solid line)

적 모델의 오차가 상대적으로 더 작음을 알 수 있고, 그 수치는 각각 0.65 (Deg)와 0.31 (Deg) 이내이다. 이때, 평균 절대 비율 오차의 값의 경우, 100%를 넘는 수치는 [Table 4]의 수식처럼



[Fig. 10] Comparison of model-learning errors based on RNN algorithm (red dotted line) and MLPNN algorithm (blue solid line)

[Table 7] Model-learning errors based on MLPNN algorithm

Joint No.	MAE (Deg)	RMSE (Deg)	MAPE (%)
1	0.3271	0.3777	292.2118
2	0.4906	0.5655	132.0476
3	0.3598	0.4154	480.9790
4	0.5397	0.6232	158.0538
5	0.3435	0.3966	30.4942
6	0.3762	0.4343	130.2757
7	0.4089	0.4721	7.9368

[Table 8] Model-learning errors based on RNN algorithm

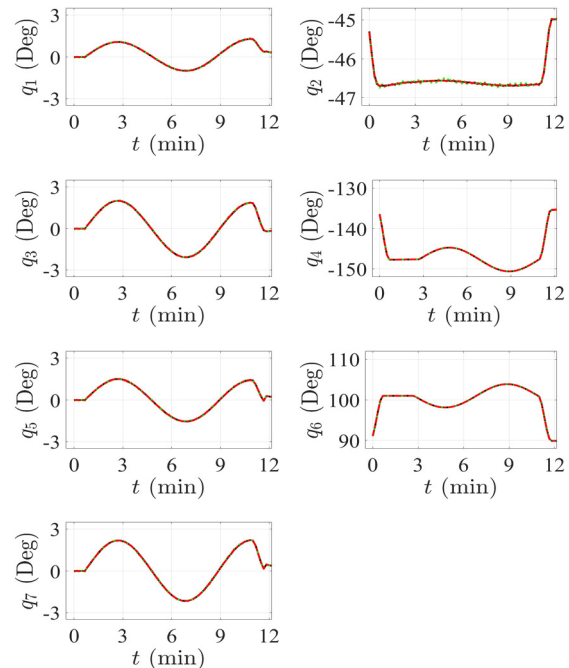
Joint No.	MAE (Deg)	RMSE (Deg)	MAPE (%)
1	0.1636	0.1888	96.0540
2	0.2453	0.2833	16.2495
3	0.2126	0.2455	243.3566
4	0.2617	0.3021	25.0932
5	0.1799	0.2077	63.5997
6	0.2290	0.2644	40.1999
7	0.1963	0.2266	55.8112

$|y_j - \hat{y}_j|$ 이 y_j 보다 크기 때문이며, 이 수학적 의미는 100%에 가까운 값을 가질수록 더 나은 성능을 제공함을 암시한다.

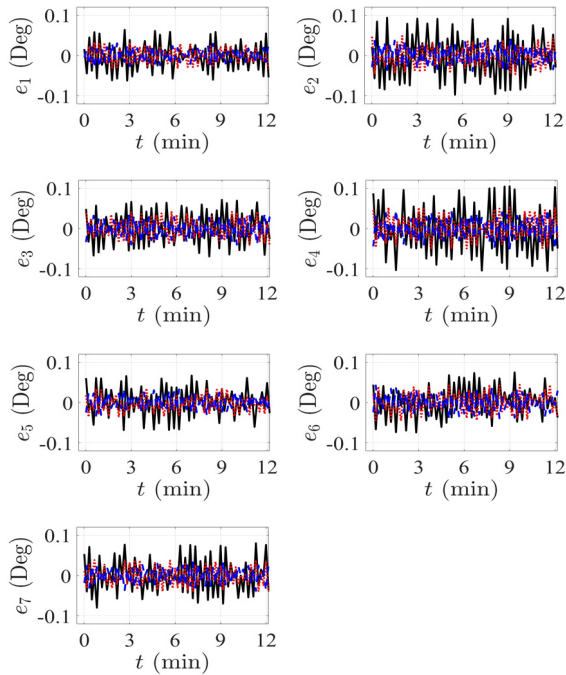
[Fig. 11]은 두 신경망 알고리즘 기반 학습된 동적 모델에 근간을 둔 페루프 제어 시뮬레이터를 이용하여 각 관절에서 도출된 각도의 움직임을 나타낸다. [Fig. 11]에서 볼 수 있듯이, 두 알고리즘에 대해 각 관절 각도에서의 움직임을 덧붙여 표현하였으며, 페루프 제어 시뮬레이터에서의 목표 경로에 대한 추종 오차는 [Fig. 12]에 도식적으로 표현하였다.

[Fig. 12]에서는 두 신경망 알고리즘에 학습된 동적 모델의 추종 오차와 가상의 로봇 매니플레이터에서의 추종 오차와의 유사도를 증점적으로 확인해야 한다. 다시 말해, 순환 신경망 알고리즘에 학습된 동적 모델에 근간을 둔 추종 오차가 가상의 로봇 매니플레이터에서의 추종 오차와 가장 유사함을 가지적으로 확인할 수 있으며, 정상적인 추종 오차는 [Table 9]-[Table 11]에 명시하였다.

[Table 9]와 [Table 10]에서 나타난 바와 같이 순환 신경망 알고리즘에 학습된 동적 모델과 다층 퍼셉트론 신경망 알고리즘에 학습된 동적 모델에서의 목표 경로에 대한 추종 오차는 평균 절대 오차와 제곱 평균 제곱근 오차에서 각각 0.07 (Deg)와 0.04 (Deg) 이내의 값을 보인다. 이 결과는 다층 퍼셉트론 신경망 알고리즘에 학습된 동적 모델보다 순환 신경망 알고리즘에 학습된 동적 모델이 [Table 11]에 나타난 가상의 로봇 매니플레



[Fig. 11] Comparison of tracking trajectories in closed-loop control simulation with system models based on RNN algorithm (red dotted line) and MLPNN algorithm (blue dashed line) on reference trajectories (black solid line)



[Fig. 12] Comparison of tracking performance in closed-loop control simulation with a virtual robot manipulator (blue dashed line) and system models based on RNN algorithm (red dotted line) and MLPNN algorithm (black solid line)

[Table 9] Tracking errors in closed-loop control simulation based on MLPNN algorithm

Joint No.	MAE (Deg)	RMSE (Deg)	MAPE (%)
1	0.0333	0.0363	62.3053
2	0.0512	0.0556	77.1851
3	0.0374	0.0398	44.3357
4	0.0565	0.0613	74.6170
5	0.0328	0.0367	93.5682
6	0.0364	0.0453	75.9809
7	0.0407	0.0491	90.4251

[Table 10] Tracking errors in closed-loop control simulation based on RNN algorithm

Joint No.	MAE (Deg)	RMSE (Deg)	MAPE (%)
1	0.0155	0.0191	80.1661
2	0.0234	0.0294	87.9360
3	0.0203	0.0253	64.6154
4	0.0247	0.0315	86.9565
5	0.0193	0.0198	96.5300
6	0.0218	0.0254	86.5323
7	0.0165	0.0214	95.8268

[Table 11] Tracking errors in closed-loop control simulation based on a virtual robot manipulator

Joint No.	MAE (Deg)	RMSE (Deg)	MAPE (%)
1	0.0154	0.0177	80.4572
2	0.0236	0.0264	87.5619
3	0.0203	0.0231	65.4999
4	0.0252	0.0284	86.0047
5	0.0180	0.0195	95.8082
6	0.0220	0.0249	86.1656
7	0.0187	0.0213	95.2572

이러로부터 수집된 데이터와 유사함을 나타내는 지표라 할 수 있다. 따라서, 다층 퍼셉트론 신경망 알고리즘을 사용하는 것보다 로봇 매니퓰레이터의 동적 모델을 학습하는 데 순환 신경망 알고리즘을 사용하는 것이 더 긍정적인 효과를 제공함을 암시한다.

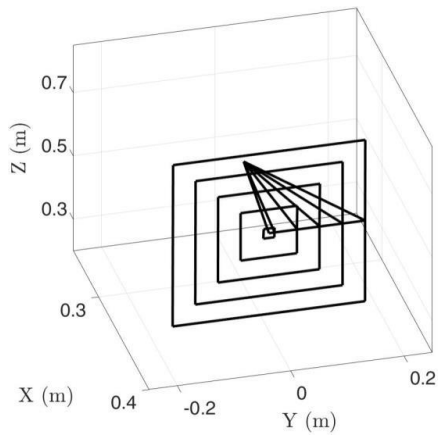
4. 토 론

4.1 실험 설명

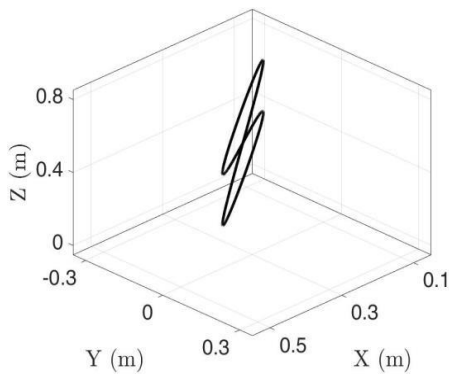
본 장에서는 3장에서 소개된 실험에 대한 이해를 돕고자 두 가지의 목적에 따른 추가적인 실험을 진행한다. 추가로 진행된 첫 번째 실험은 선정된 하이퍼 파라미터의 타당성을 판별하는데 목적을 두고 있으며, 선정된 하이퍼 파라미터가 적용되었을 때의 실험 결과를 그 외의 하이퍼 파라미터가 선정되었을 때의 실험 결과와 비교하여 선정된 하이퍼 파라미터가 최적의 설정이었음을 확인한다. 그 외의 하이퍼 파라미터들은 크게 2 가지 종류로 구성하였으며, [Table 1]에서 소개한 ① 활성화 함수를 가진 파라미터들과 ② 학습 결과에 영향을 많이 끼치는 것으로 알려진 노드 수와 은닉층 수가 서로 다른 파라미터를 가지도록 설정하였다. 이때, 노드 수와 은닉층 수의 종류는 [Table 6]에 선정된 값보다 크고 작은 값을 가지는 모델로 각각 2 개씩 선정하였으며, [Table 12]에서처럼 4 가지로 분류하여 학습 결과를 도출하였다.

[Table 12] The number of nodes by hidden layers

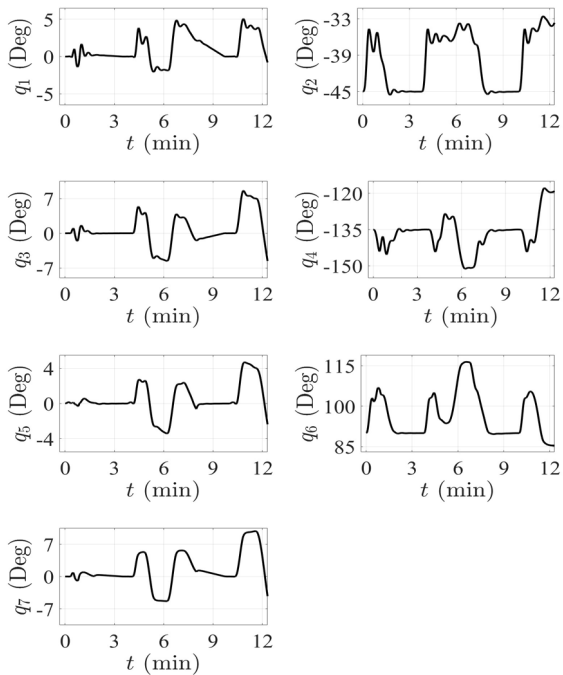
Model No.	The Number of Nodes by Hidden Layers
1	256 → 128 → 64 → 32 → 16 → 1
2	128 → 64 → 1
3	64 → 32 → 16 → 1
4	64 → 32 → 1



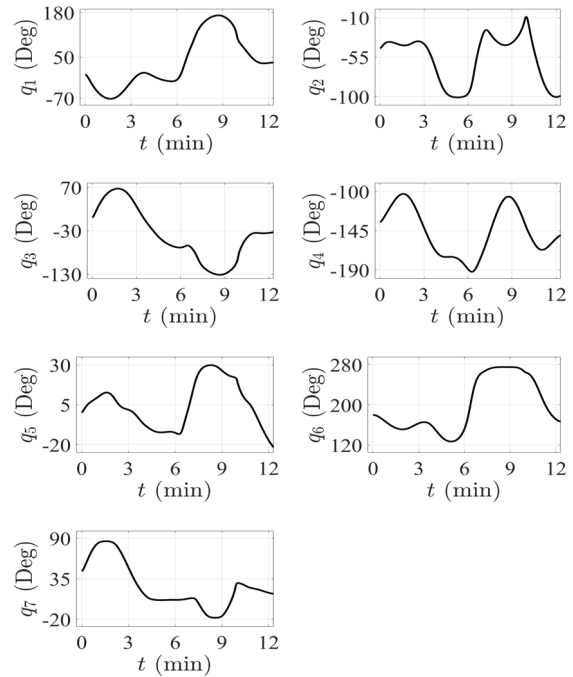
[Fig. 13] The desired position-tracking trajectories based on square-position in task space



[Fig. 14] The desired position-tracking trajectories based on number '8'-position in task space



[Fig. 15] The desired angles based on square-position in joint space



[Fig. 16] The desired angles based on number '8'-position in joint space

추가로 진행된 두 번째와 세 번째 실험은 순환 신경망 알고리즘에 학습된 동적 모델의 범용성을 검증하는 데 목적을 두고 있다. 두 번째 실험의 경우에는 [Fig. 13]에서처럼 기준 경로를 생성하는 데 경로에 대한 이동 속도와 크기가 다른 사각형 경로에 근간을 두어 진행되었다. 세 번째 실험의 경우에는 [Fig. 14]처럼 궤적에 대한 이동 속도와 크기가 다른 숫자 '8' 경로가 기준 경로를 생성하는 데 기반이 되었다. 또한, 이들은 페루프 제어 시뮬레이션에서의 목표 경로로 활용되었으며, 관절 공간에서의 기준 경로들은 자코비안 행렬을 통해 변환되어 사용하였다. [Fig. 13]과 [Fig. 14]에서 도식적으로 표현된 위치에 대한 관절 공간의 목표 경로는 [Fig. 15]와 [Fig. 16]에서 확인할 수 있다.

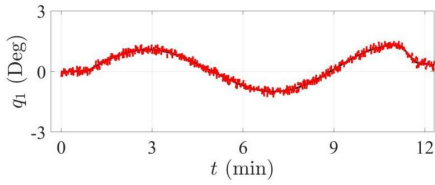
이 장에서 진행된 모든 실험은 3장에서 실험 설정과 모두 동일하게 진행되었다.

4.2 실험 결과

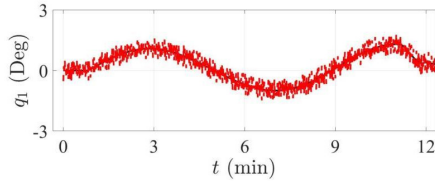
4.2.1 활성화 함수에 따른 실험 결과

[Fig. 17]~[Fig. 19]는 순환 신경망 알고리즘에 [Table 1]에서 소개된 활성화 함수를 적용하였을 때의 출력값에 대한 목표값을 덧붙여 보여주고 있다. 이에 대한 학습 오차는 [Fig. 20]~[Fig. 22]에 도식적으로 표현하였다.

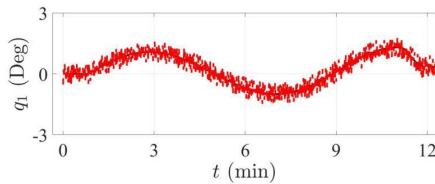
[Fig. 20]~[Fig. 22]는 [Fig. 17]~[Fig. 19]에서 도식화된 결괏값에 근간을 두어 출력값에 대한 오차를 나타내고 있다. 이때,



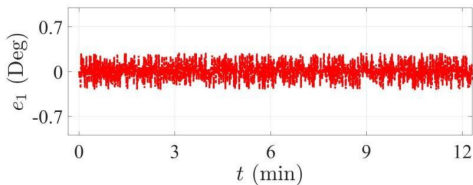
[Fig. 17] Comparison of reference trajectories (black solid line) output results (red dotted line) in accordance with RNN algorithm with hyperbolic tangent function



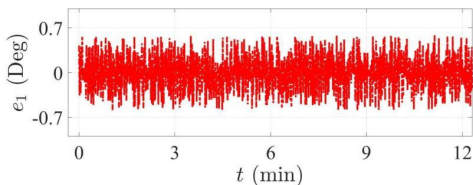
[Fig. 18] Comparison of reference trajectories (black solid line) output results (red dotted line) in accordance with RNN algorithm with sigmoid function



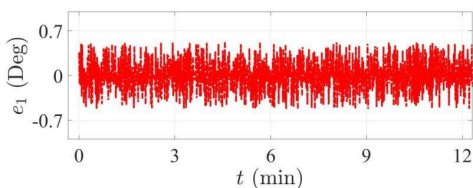
[Fig. 19] Comparison of reference trajectories (black solid line) output results (red dotted line) in accordance with RNN algorithm with ReLU function



[Fig. 20] Model-learning errors based on RNN algorithm with hyperbolic tangent function



[Fig. 21] Model-learning errors based on RNN algorithm with sigmoid function



[Fig. 22] Model-learning errors based on RNN algorithm with ReLU function

[Table 13] Model learning errors by activation functions

Activation Function	MAE (Deg)	RMSE (Deg)	MAPE (Deg)
Hyperbolic Tangent	0.15	0.17	76.53
Sigmoid	0.25	0.28	194.18
ReLU	0.28	0.32	233.33

출력된 최대 오차는 하이퍼볼릭 탄젠트, 시그모이드 그리고 렐루 함수에서 각각 약 0.3 (Deg), 약 0.5 (Deg), 약 0.6 (Deg)를 보이며, 하이퍼볼릭 탄젠트 함수를 활성화 함수로 사용하였을 때의 결과가 가장 긍정적임을 알 수 있다. 각각의 활성화 함수에 따른 순환 신경망 알고리즘의 정량적인 오차는 [Table 13]에서 나타내었다.

[Table 13]에서 확인할 수 있듯이, 하이퍼볼릭 탄젠트 함수가 활성화 함수로 사용되었을 때의 평균 절대 오차와 제곱 평균 제곱근 오차의 값이 가장 작음을 알 수 있으며, 평균 절대 비율 오차의 값도 100%에 가장 가까운 값을 가짐을 확인할 수 있다.

4.2.2 노드 및 은닉층 수에 따른 실험 결과

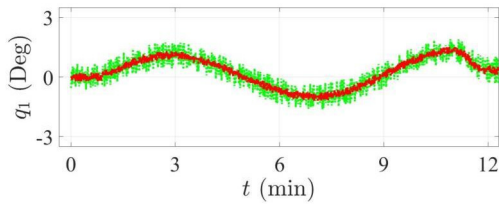
[Fig. 23]~[Fig. 26]은 [Table 12]에 나타낸 각각의 모델의 실험 결과로서, 순환 신경망 알고리즘의 출력값에 목표값을 덧붙여 표현하였다. 이때, [Fig. 21]~[Fig. 24]의 순서는 [Table 12]에 나타낸 모델의 순서와 동일하게 나타내었으며, 이에 대한 학습 오차는 [Fig. 23]~[Fig. 26]에 도식적으로 나타내었다.

[Fig. 27]~[Fig. 30]에 나타낸 바와 같이, [Table 12]에 나타낸 각각의 모델에 대한 학습 오차의 최댓값은 [Table 12]에 나타낸 모델의 순서대로 약 0.7 (Deg), 약 0.4 (Deg), 약 0.6 (Deg), 약 1.0 (Deg)를 보이고 있다. 이때, 두 번째 모델에서의 학습 오차가 비교군 모델 중에서는 가장 작지만, 본 논문에서 최적의 하이퍼파라미터로 설정된 [Fig. 20]의 결과와 비교하면 약 1.33 배 크음을 알 수 있다.

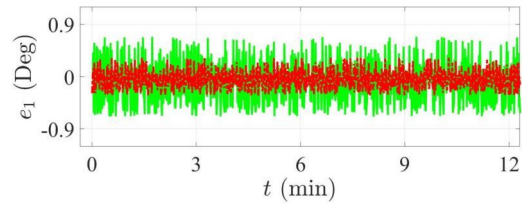
4.2.3 비(非)학습 데이터 활용 페루프 제어 실험 결과

[Fig. 31]은 순환 신경망 알고리즘에 학습된 동적 모델에 근간을 둔 페루프 제어 시뮬레이터를 이용하여 각 관절에 대해 도출된 각도의 움직임을 목표 경로와 함께 나타내었다. 이때, 페루프 제어 시뮬레이터에서의 목표 경로에 대한 추종 오차는 [Fig. 32]에 도식적으로 표현하였다.

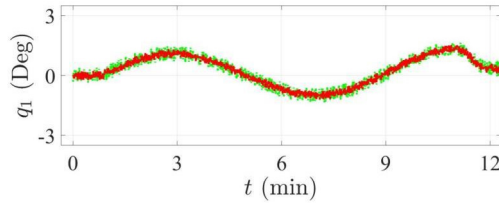
[Table 14]는 순환 신경망 알고리즘에 학습된 동적 모델이 로봇 매니퓰레이터가 작업 공간 내 사각형 경로를 추종할 때의 각도 오차를 측정 지표들을 통해 나타내고 있다. 이 수치는 앞서 [Table 10]~[Table 11]과의 비교를 통해 유사함을 나타내는 지표라 할 수 있다. 이를 통해 학습에 사용된 원형 경로 설정 방법은 학습된 순환 신경망 알고리즘 기반 동적 모델이 범용성을



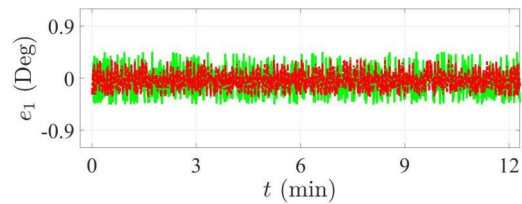
[Fig. 23] Comparison of reference trajectories (red solid line) output results (green dotted line) on model #1 in [Table 12]



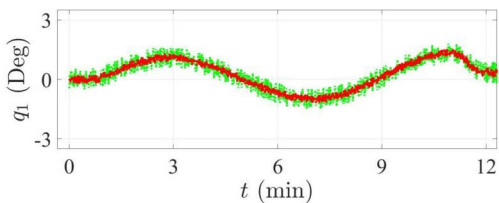
[Fig. 27] Comparison of model-learning errors between the proposed hyper-parameter and hyper-parameter of model #1



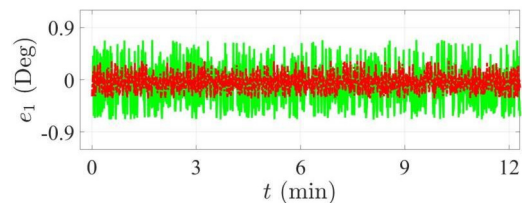
[Fig. 24] Comparison of reference trajectories (red solid line) output results (green dotted line) on model #2 in [Table 12]



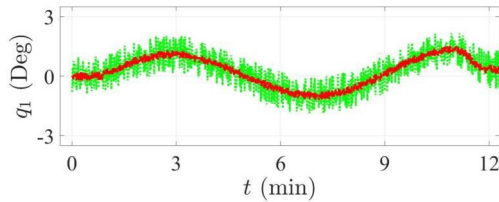
[Fig. 28] Comparison of model-learning errors between the proposed hyper-parameter and hyper-parameter of model #2



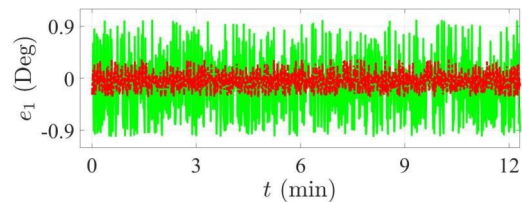
[Fig. 25] Comparison of reference trajectories (red solid line) output results (green dotted line) on model #3 in [Table 12]



[Fig. 29] Comparison of model-learning errors between the proposed hyper-parameter and hyper-parameter of model #3



[Fig. 26] Comparison of reference trajectories (red solid line) output results (green dotted line) on model #4 in [Table 12]



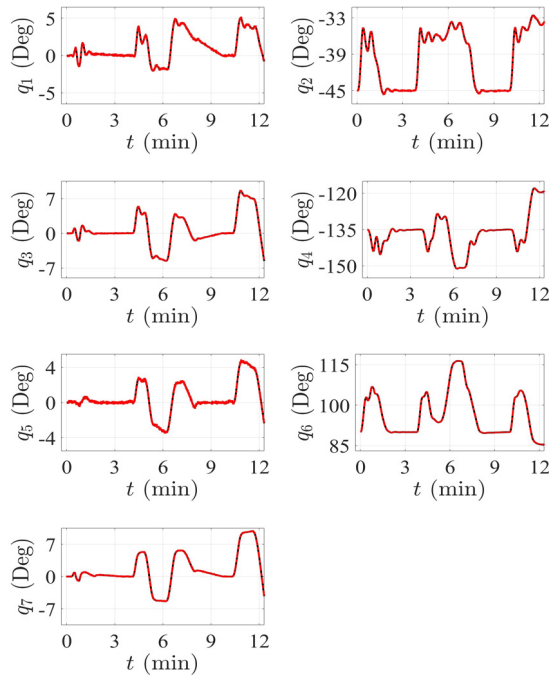
[Fig. 30] Comparison of model-learning errors between the proposed hyper-parameter and hyper-parameter of model #4

가질 수 있다는 긍정적인 결과를 의미함을 알 수 있다.

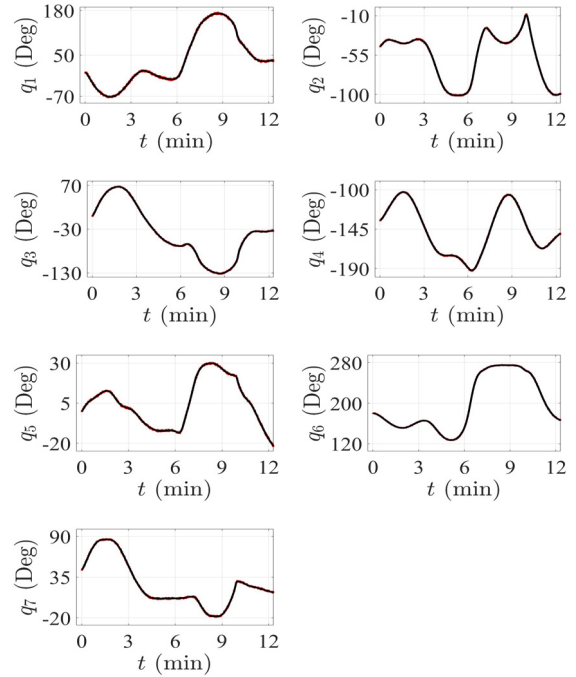
[Fig. 33]은 순환 신경망 알고리즘에 학습된 동적 모델에 근간을 둔 페루프 제어 시뮬레이터를 이용하여 각 관절에서 도출된 각도를 목표 경로와 함께 나타내었다. 이때, 학습에 사용된 원형 경로와 숫자‘8’형태의 경로에 대한 추종 오차의 절대치는 큰 차이가 없으나, 숫자‘8’형태의 경로에 대한 오차는 원형 경로를 추종할 때의 오차보다 상대적으로 성능이 저하되었음을 확인할 수 있다. 이와 같은 결과가 발생한 원인은 숫자‘8’형태의 경로가 x축, y축 그리고 z축에서의 변칙적인 경로를 내포하기 때문이다. 페루프 제어 시뮬레이터에서의 목표 경로에 대한 추종 오차는 [Fig. 34]에 도식적으로 표현하였으며, 그 수치는 [Table 15]에서 확인할 수 있다.

5. 결 론

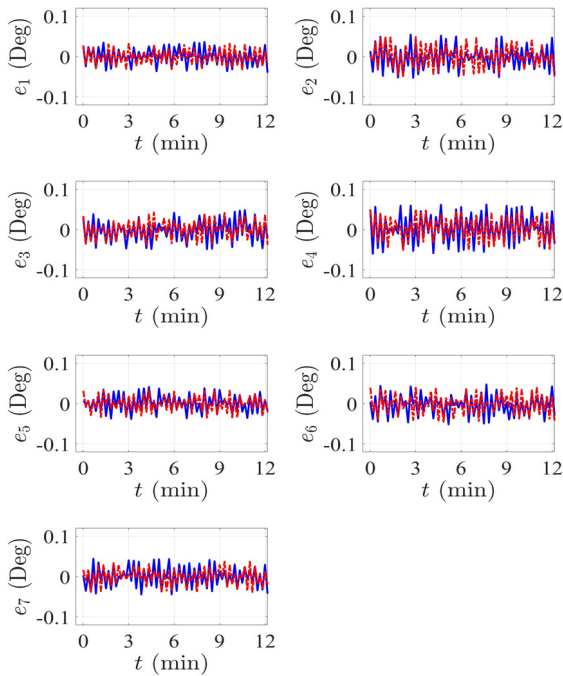
본 논문은 다관절 로봇 매니플레이터의 동적 모델을 학습하는 데 순환 신경망 알고리즘에 근간을 두어 제안하였다. 제안된 학습 방법에서 로봇 매니플레이터로부터 측정된 축별 각도, 각속도 그리고 토크에 대한 정보는 순환 신경망 알고리즘의 입력 데이터로 사용되었으며, 축별 각도만을 순환 신경망 알고리즘의 출력 데이터로 적용되었다. 또한, 로봇 매니플레이터의 동적 모델을 학습하는 데 최적의 하이퍼 파라미터를 선정하였으며, 여러 가지 하이퍼 파라미터와의 비교를 통해 선정된 하이퍼 파라미터에 대한 타당성을 검증하였다. 추가적으로, 학습된



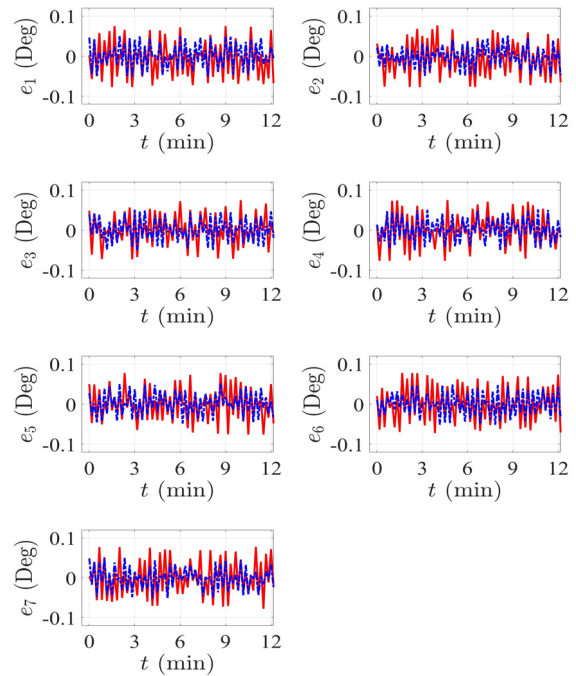
[Fig. 31] Comparison of tracking trajectories in closed-loop control simulation with system models based on RNN algorithm (red dotted line) on reference trajectories (black solid line)



[Fig. 33] Comparison of tracking trajectories in closed-loop control simulation with system models based on RNN algorithm (red dotted line) on reference trajectories (black solid line)



[Fig. 32] Comparison of circular trajectory-tracking errors (blue dashed line) and square trajectory-tracking errors (red dotted line) in closed-loop control simulation with system models based on RNN algorithm



[Fig. 34] Comparison of circular trajectory-tracking errors (blue dashed line) and number '8'-shaped trajectory-tracking errors (red dotted line) in closed-loop control simulation with system models based on RNN algorithm

[Table 14] Square trajectory-tracking errors in closed-loop control simulation based on RNN algorithm

Joint No.	MAE (Deg)	RMSE (Deg)	MAPE (%)
1	0.0157	0.0193	80.2187
2	0.0238	0.0297	87.9867
3	0.0201	0.0251	64.4136
4	0.0244	0.0312	86.9897
5	0.0195	0.0121	96.8437
6	0.0221	0.0256	86.9532
7	0.0163	0.0216	96.0126

[Table 15] Number '8'-shaped trajectory-tracking errors in closed-loop control simulation based on RNN algorithm

Joint No.	MAE (Deg)	RMSE (Deg)	MAPE (%)
1	0.0393	0.0401	78.3879
2	0.0397	0.0421	84.3199
3	0.0376	0.0403	67.4239
4	0.0373	0.0431	81.5423
5	0.0435	0.0423	93.1453
6	0.0386	0.0443	84.1013
7	0.0411	0.0429	89.3543

동적 모델을 전달함수로 둔 피드백 제어 시뮬레이션을 진행하여 제안된 학습 과정으로부터 얻어진 동적 모델에 대한 성능을 분석하였으며, 학습된 동적 모델에 대한 정확도는 가상의 로봇 매니플레이터에서의 페루프 제어 시뮬레이션에서 얻은 축별 각도에 대한 추적 성능과 비교하여 유사함을 입증하였다.

References

- [1] J. Qin, F. Léonard, and G. Abba, "Real-time Trajectory Compensation in Robotic Friction Stir Welding Using State Estimators," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 6, pp. 2207-2214, Nov., 2016, DOI: 10.1109/TCST.2016.2536482.
- [2] H. Park, J. Park, D.-H. Lee, J.-H. Park, M.-H. Baeg, and J.-H. Bae, "Compliance-based Robotic Peg-in-hole Assembly Strategy Without Force Feedback," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 8, pp. 6299-6309, Aug., 2017, DOI: 10.1109/TIE.2017.2682002.
- [3] X. Yu, B. Li, W. He, Y. Feng, L. Cheng, and C. Silvestre, "Adaptive-constrained Impedance Control for Human-robot Co-transportation," *IEEE Transactions on Cybernetics*, vol. 52, no. 12, pp. 13237-13249, Dec., 2022, DOI: 10.1109/TCYB.2021.3107357.
- [4] F. Santoso and A. Finn, "Trusted Operations of a Military Ground Robot in the Face of Man-in-the-Middle Cyber-Attacks Using Deep Learning Convolutional Neural Networks: Real-Time Experimental Outcomes," *IEEE Transactions on Dependable and Secure Computing*, pp. 1-12, Aug., 2023, DOI: 10.1109/TDSC.2023.3302807.
- [5] M. Cai, Y. Wang, S. Wang, R. Wang, Y. Ren, and M. Tan, "Grasping Marine Products with Hybrid-driven Underwater Vehicle-manipulator System," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 3, pp. 1443-1454, Jul., 2020, DOI: 10.1109/TASE.2019.2957782.
- [6] F. Aghili, "Robust Impedance-matching of Manipulators Interacting with Uncertain Environments: Application to Task Verification of the Space Station's Dexterous Manipulator," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 4, pp. 1565-1576, Aug., 2019, DOI:10.1109/TMECH.2019.2928281.
- [7] J. Baek and M. Kang, "A Synthesized Sliding-mode Control for Attitude Trajectory Tracking of Quadrotor UAV Systems," *IEEE/ASME Transactions On Mechatronics*, vol. 28, no. 4, pp. 2189-2199, Aug., 2023, DOI: 10.1109/TMECH.2022.3230755.
- [8] M. Kobayashi and N. Motoi, "BSL: Navigation Method Considering Blind Spots Based on ROS Navigation Stack and Blind Spots Layer for Mobile Robot," *IEEE Transactions on Industry Applications*, vol. 60, no. 1, pp. 1695-1704, Jan.-Feb., 2024, DOI: 10.1109/TIA.2023.3312649.
- [9] H.-C. Huang and Y.-X. Chen, "Evolutionary Optimization of Fuzzy Reinforcement Learning and its Application to Time-Varying Tracking Control of Industrial Parallel Robotic Manipulators," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 12, pp. 11712-11720, Dec., 2023, DOI: 10.1109/TII.2023.3248120.
- [10] W. Li, X. Huang, L. Yan, H. Cheng, B. Liang, and W. Xu, "Force Sensing and Compliance Control for a Cable-driven Redundant Manipulator," *IEEE/ASME Transactions on Mechatronics*, vol. 29, no. 1, pp. 777-788, Feb., 2024, DOI: 10.1109/TMECH.2023.3263922.
- [11] Z. Xie, S. Li, and L. Jin, "A Bi-criteria Kinematic Strategy for Motion/Force Control of Robotic Manipulator," *IEEE Transactions on Automation Science and Engineering*, 2023, DOI: 10.1109/TASE.2023.3313564.
- [12] J. Baek, M. Jin, and S. Han, "A New Adaptive Sliding-mode Control Scheme for Application to Robot Manipulators," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 6, pp. 3628-3637, Jun., 2016, DOI: 10.1109/TIE.2016.2522386.
- [13] J. Baek, S. Cho, and S. Han, "Practical Time-delay Control with Adaptive Gains for Trajectory Tracking of Robot Manipulators," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5682-5692, Jul., 2018, DOI: 10.1109/TIE.2017.2782238.
- [14] J. Baek, W. Kwon, B. Kim, and S. Han, "A Widely Adaptive Time-delayed Control and its Application to Robot Manipulators," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 7, pp. 5332-5342, Jul., 2019, DOI: 10.1109/TIE.2018.2869347.
- [15] J. Baek, W. Kwon, and C. Kang, "A New Widely and Stably

- Adaptive Sliding-mode Control with Nonsingular Terminal Sliding Variable for Robot Manipulators,” *IEEE Access*, vol. 8, pp. 43443-43454, 2020, DOI: 10.1109/ACCESS.2020.2977434.
- [16] J. Duan, W. Cao, Y. Zheng, and L. Zhao, “On the Optimization Landscape of Dynamic Output Feedback Linear Quadratic Control,” *IEEE Transactions on Automatic Control*, vol. 69, no. 2, pp. 920-935, Feb., 2024, DOI: 10.1109/TAC.2023.3275732.
- [17] M. Makarov, M. Grossard, P. Rodriguez-Ayerbe, and D. Dumur, “Modeling and Preview H_∞ Control Design for Motion Control of Elastic-joint Robots with Uncertainties,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 10, pp. 6429-6438, Oct., 2016, DOI: 10.1109/TIE.2016.2583406.
- [18] R. H. Middleton and G. C. Goodwin, “Adaptive Computed Torque Control for Rigid Link Manipulators,” *1986 25th IEEE Conference on Decision and Control*, Athens, Greece, pp. 68-73, 1986, DOI: 10.1109/CDC.1986.267156.
- [19] L. Almeida, 1997, Multilayer Perceptrons, *Handbook of Neural Computation*, [Online], <http://www.lx.it.pt/~lbalmeida/papers/AlmeidaHNC.pdf>.
- [20] D. Cagigas-Muñiz, “Artificial Neural Networks for Inverse Kinematics Problem in Articulated Robots,” *Engineering Applications of Artificial Intelligence*, vol. 126, Nov., 2023, DOI: 10.1016/j.engappai.2023.107175.
- [21] M. Defdaf, F. Berrabah, A. Chebabhi, and B. D. E. Cherif, “A New Transform Discrete Wavelet Technique Based on Artificial Neural Network for Induction Motor Broken Rotor Bar Faults Diagnosis,” *International Transactions on Electrical Energy Systems*, vol. 31, no. 4, pp. e12807, Apr., 2021, DOI: 10.1002/2050-7038.12807.
- [22] A. Eskandarian, N. E. Bedewi, B. M. Kramer, and A. J. Barbera, “Dynamics Modeling of Robotic Manipulators Using An Artificial Neural Network,” *Journal of Robotic Systems*, vol. 11, no. 1, pp. 41-56, Apr., 1993, DOI: 10.1002/rob.4620110106.
- [23] M. Schuster and K. K. Paliwal, “Bidirectional Recurrent Neural Networks,” *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673-2681, Nov., 1997, DOI: 10.1109/78.650093.
- [24] L. Jin, M. M. Gupta, and P. N. Nikiforuk, “Dynamic Recurrent Neural Networks for Modeling Flexible Robot Dynamics,” *Proceedings of the 10th IEEE International Symposium on Intelligent Control*, Monterey, CA, USA, pp. 105-110, 1995, DOI: 10.1109/ISIC.1995.525045.
- [25] Y. Bengio, P. Frasconi, P, and P. Simard, “The Problem of Learning Long-term Dependencies in Recurrent Networks,” *In IEEE International Conference on Neural Networks*, San Francisco, USA, pp. 1183-1188, 1993, DOI: 10.1109/ICNN.1993.298725.
- [26] M. Hüskens and P. Stagge, “Recurrent Neural Networks for Time Series Classification,” *Neurocomputing*, vol. 50, pp. 223-235, Jan., 2003, DOI: 10.1016/S0925-2312(01)00706-8.
- [27] A. De Myttenaere, B. Golden, B. Le Grand, and F. Rossi, “Mean Absolute Percentage Error for Regression Models,” *Neurocomputing*, vol. 192, pp. 38-48, Jun., 2016, DOI: 10.1016/j.neucom.2015.12.114.
- [28] G. Cohen, S. Afshar, G. Orchard, J. Tapson, R. Benosman, and A. van Schaik, “Spatial and Temporal Downsampling in Event-based Visual Classification,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, pp. 5030-5044, Oct., 2018, DOI: 10.1109/TNNLS.2017.2785272.
- [29] Y. Yu, T. Zhi, X. Zhou, S. Liu, Y. Chen, and S. Cheng, “Bshift: a Low Cost Deep Neural Networks Accelerator,” *International Journal of Parallel Programming*, vol. 47, pp. 360-372, Jan., 2019, DOI: 10.1007/s10766-018-00624-9.
- [30] D. Masters and C. Luschi, “Revisiting Small Batch Training for Deep Neural Networks,” *arXiv:1804.07612*, 2018, DOI: 10.48550/arXiv.1804.07612.
- [31] J. Bergstra, and Y. Bengio, “Random search for hyper-parameter optimization,” *Journal of machine learning research*, vol. 13, no. 2, pp. 281-305, Feb., 2012, [Online], <https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf?ref=broutonlab.com>.
- [32] L. Yang and A. Shami, “On hyperparameter optimization of machine learning algorithms: Theory and practice,” *Neurocomputing*, vol. 415, pp. 295-316, Nov., 2020, DOI: 10.1016/j.neucom.2020.07.061.
- [33] M. W. Spong, S. Hutchinson, and M. Vidyasagar, “Dynamics,” *Robot modeling and control*, 2th ed. John Wiley & Sons, 2020, ch. 6, sec. 6.6, pp. 198-208, [Online], https://books.google.co.kr/books?hl=ko&lr=&id=DdjNDwAAQBAJ&oi=fnd&pg=PR2&dq=M.+W.,+Spong,+S.,+Hutchinson,+and+M.,+Vidyasagar,+%E2%80%9CDynamics,%E2%80%9D+Robot+modeling+and+control,+2th+ed.+John+Wiley+%26+Sons,+2020,+ch.+6,+sec.+6.6,+pp.+198-208.&ots=Cy1s_TEGtG&sig=BjegS9gBp8F_gjqhzzMEKxBPMqU&redir_esc=y#v=onepage&q&f=false.



신 승 천

2024 국립강릉원주대학교 기계공학과
(공학사)

2024~현재 국립강릉원주대학교 기계공학과
(석사과정)

관심분야: Robot control, Embedded system, AI, Learned Dynamics Model



강 동 엽

2006 포항공과대학교 전자전기공학과(공학사)

2008 포항공과대학교 친환경소재대학원
(공학석사)

2013 포항공과대학교 친환경소재대학원
(공학박사)

2008~2014 포스코 책임연구원

2015~현재 한국전자통신연구원 책임연구원

관심분야: Automatic control, Control theory, Nonlinear systems, Robotics



백 재 민

2012 고려대학교 기계공학과(공학사)

2018 포항공과대학교 IT융합공학과
(공학박사)

2018~2020 국방과학연구소 선임연구원

2020~현재 국립강릉원주대학교 기계공학과
부교수

관심분야: Adaptive/Robust control, Control theory, Robot control, Artificial intelligence (AI)